Hindawi
Journal of Electrical and Computer Engineering
Volume 2023, Article ID 3615137, 26 pages
https://doi.org/10.1155/2023/3615137

Research Article

Investigation on Machine Learning Approaches for Environmental Noise Classifications

Ali Othman Albaji, Rozeha Bt. A. Rashid, and Siti Zeleha Abdul Hamid

Telecommunication Software and Systems (TeSS) Research Group, Department of Communication Engineering, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Johor Bahru, Malaysia

Correspondence should be addressed to Ali Othman Albaji; sabos_in@yahoo.com and Rozeha Bt. A. Rashid; rozeha@utm.my

Received 5 October 2022; Revised 10 March 2023; Accepted 19 April 2023; Published 31 May 2023

Academic Editor: Ping-Feng Pai

This project aims to investigate the best machine learning (ML) algorithm for classifying sounds originating from the environment that were considered noise pollution in smart cities. Sound collection was carried out using necessary sound capture tools, after which ML classification models were utilized for sound recognition. Additionally, noise pollution monitoring using Python was conducted to provide accurate results for sixteen different types of noise that were collected in sixteen cities in Malaysia. The numbers on the diagonal represent the correctly classified noises from the test set. Using these correlation matrices, the F1 score was calculated, and a comparison was performed for all models. The best model was found to be random forest.

1. Introduction

Noise pollution is one of the most significant problems of the modern world, caused by various sources such as industrial noise, road noise, work noise, and human conversation. A thorough study was conducted to predict the type of noise that occurs based on a set of specific predictors. This project aims to predict the frequency with which different types of noise occur in Malaysia. To achieve this goal, a dataset of 13 different features was used to predict the output column. Exploratory data analysis was conducted to obtain an overview of the data, followed by necessary data preprocessing steps to feed the data to different supervised learning algorithms. The methodology adopted for noise classification will be explained in this research. The approach used in this study is a comparative analysis of five machine learning algorithms, including decision tree “DT,” random forest “RF,” logistic regression “LR,” K-nearest neighbor “KNN,” and support vector machine “SVM.” These algorithms have demonstrated excellent results in various fields, which is why all of them were utilized to obtain the best model. The F1 score was used as the primary accuracy measure as it represents a balance between recall and precision. The objective of this study is to develop a machine learning technique that can classify noise levels. A dataset of 873 audio samples was used to evaluate the effectiveness of the technique, and the noise levels were tested against WHO standards. The dataset consists of 873 sounds that were captured from various environments using https://monitornoises.com [1], and the samples were categorized into 16 different classes, including indoor, chatting, road, industrial, wind, footsteps, and others. The major contributions of this paper are as follows:

(i) Create a classifier using machine learning technique that can categorize environmental sounds and soundscape map that can assist relevant agencies and industries in decision and policy making to mitigate noise pollution

(ii) Investigate possible machine learning algorithms such as KNN, RF, SVM, DT, and LR for the sound classifier

(iii) Evaluate the approach using a recorded dataset of 873 samples of environmental sounds

2. Previous Work

As per Marjanovic et al., their research focuses on developing detailed maps of pollutants and noise to identify urban areas that affect human health. They demonstrate how a comprehensive framework is established, including sensor calibration, data collection, and processing. Noise pollution can cause negative health effects such as hearing loss, stress, blood pressure fluctuations, migraines, sleeplessness, nervous system disorders, productivity restrictions, and mental health issues. To detect noise pollution and monitor its impact on human health, a system combining a wireless sensor network (WSN) with a body access network (BAN) was developed. The WSN and BAN networks allow scientists to study the hazards of excessive noise and its effects on people's health [2].

A new technology has been developed by Kulkarni et al. that can detect hazardous chemicals and loud noises. This technology is a novel concept that can also detect air and sound pollution. The main purpose of this system is to monitor the environment using two sensors, the MQ135 and a microphone sound sensor. The MQ135 sensor is used to detect NH_3, CO_2, SO_2, and other dangerous gases and sends a signal to the control unit when a situation is detected [3].

Patil has presented Internet of Things (IoT)-based solutions to address these issues. The system monitors vehicle pollution and noise levels, and any readings that exceed a set limit are immediately reported to the relevant authorities, including the traffic department and environmental organizations [4].

A smart city has been suggested by Sumithra et al. Using sensors and modules, they were able to monitor a wide range of environmental conditions. The data are monitored and sent to the cloud server using air and sound sensors. Cloud storage is responsible for storing and analyzing the data that is collected [5].

Zimmerman and Robson have developed a noise model and data visualization tools to assist prospective homebuyers. The nighttime noise of residents is represented as an ambient sound source, and the researchers aim to assess its impact. The final stage of the implementation attempts to alert users via SMS. The system has also been enhanced with additional features. A quantitative study of the noise environment is provided by the noise analysis output supplied through SMS text messages when the peak value is reached. The noise model has been implemented via the creation, calibration, and verification of a device. The physical propagation model is the first approach for predicting noise from a distance, using the distance between the sound source and the predicted noise location point as well as the physical qualities of the sound source. Several industrialized nations have been doing extensive research on airport noise control since the 1960s, including the FAA's integrated noise model (INM), UK's aircraft noise contour (ANCON) model, and Switzerland's Fluglaerm (FLULA) program. Predicting ambient noise has necessitated research on the spatial propagation of noise. It is possible to design a noise distribution by using the propagation model of noise attenuation [6]. There have also been investigations on the most effective locations for noise monitoring stations [5]. Physical propagation models are seldom used in investigations looking at the temporal variance of urban noise.

The factual methodology is the second type of noise prediction strategy. Kumar and Jain [7] proposed an autoregressive integrated moving average (ARIMA) model for traffic noise time series prediction, even after just a few hours of data. A larger time horizon is shown to be ideal when examining the time series. Kumar and Jain [7] conducted an analysis of long-term data from noise monitoring and showed that the ARIMA approach is reliable for time series modeling of traffic noise although different scenarios require adjusting underlying parameters.

According to Prieto Gajardo et al. [8], a Fourier analysis of traffic noise in the cities of Cáceres and Talca showed that larger seasonal components and amplitude values are consistent in different samples regardless of the city's measured weather and associated traffic flow, indicating that urban traffic noise can be predicted. Regression models have also been used to predict the noise level in specific industrial conditions by determining the dominant frequency limit.

The machine learning approach is categorized as a noise prediction method. To simulate traffic noise, a back-propagation neural network is used. Prieto Gajardo et al. [8] used a random forest regression approach to predict wind turbine noise. Torija and Ruiz [9] were able to accurately estimate the level of ambient noise using feature extraction and machine learning approaches. However, arranging the data is challenging due to the subtle differences between the 32 input variables.

Several factors are considered in predicting ambient noise, including historical and real-time monitoring data. Van Den Berg et al. [10] found that integrating the rules or patterns extracted from monitoring data with the acoustic theoretical calculation model improves the prediction accuracy of noise significantly. To better represent regional noise levels and increase data-gathering efficiency, certain sampling procedures have been proposed while considering the concerns of conserving resources. Based on the time savings and the accuracy of the findings, Zambon et al. [11] have observed that a noncontinuous 5- to 7-day noise observation is adequate for long-term noise prediction.

Controlling urban noise can be based on a scientific foundation if the temporal volatility of noise can be predicted. In recent years, the exponential growth of ambient noise data has been made possible due to advancements in sound level meters and sensor networks. Although there have been studies on noise measurement, prediction, and control in the past, most of the data collected has been small. With the newfound motivation to reconsider environmental noise prediction, there is a need to explore better models and methods for processing large amounts of noise data. A more efficient method for predicting noise in the temporal domain is required. However, few studies have investigated and predicted the variation in noise over a single day [11].

In recent years, there has been a rapid development of deep learning, which has proven effective in a wide range of fields [12]. Deep architectures or multiple-layer structures can identify a vast number of structures in a dataset [2].

Deep learning has been developed as a result of artificial neural networks research. The most prevalent models of neural networks are multiple-layer perceptron, convolutional neural networks, and recurrent neural networks. [13] The use of RNNs in time series analysis is a common approach for representing hidden states and capturing data characteristics. However, simple RNNs suffer from the issue of long-term dependencies and are not efficient in utilizing past information collected over extended periods. To address this problem, LSTM networks have been developed and applied to various applications such as stock price prediction, air quality monitoring, sea surface temperature monitoring, flight passenger count monitoring, and automatic speech recognition. The model's effectiveness has been demonstrated through its good performance in the results [13].

Almehmadi [14] proposed a smart city architecture that utilizes IoT technology to mitigate noise pollution by designing noise detection end nodes as a component of the noise pollution monitoring system (NPMS). This system is able to locate the area, time, level of noise, and any related events, to evaluate the proposed architecture. Additionally, Zamora et al. [15] proposed using smartphones as environmental noise-sensing devices. Therefore, they focused their study on analyzing the impact of three different noise calculation algorithms using various types of smartphones and determining their accuracy when compared to a professional noise measurement device.

In addition to the studies discussed previously, de Souza et al. [16] conducted a study comparing quantitative and qualitative results at university campuses. They conducted on-site sound measurements at the Federal University of Juiz de Fora (UFJF) in Brazil and distributed questionnaires to 140 volunteers. Their analysis revealed that the noise levels at the campus did not meet national and international regulations for an educational area.

Moreover, Wessels and Basten [17] designed five aspects of acoustic sensor networks for environmental noise monitoring: hardware costs, scalability, flexibility, reliability, and accuracy. These aspects led the researcher to create four categories that can contribute to the field of noise pollution monitoring by addressing some of the challenges.

Additionally, Noriega-Linares and Navarro Ruiz [18] have developed a low-cost sound sensor prototype based on the Raspberry Pi platform for analyzing ambient noise. The device is connected to the cloud for real-time sharing of results. Tests have demonstrated that the Raspberry Pi is a powerful and cost-effective processing core for low-cost devices. Moreover, other researchers have combined the power of smartphones to detect noise pollution [19, 20] instead of relying on an independent endpoint.

Alsouda et al. [21] presented a machine-learning framework in their paper that can classify urban noise using a low-cost IoT device. They used a combination of supervised and unsupervised methods, such as KNN and SVM, to extract audio features. The researchers collected 3,000 sound samples and tested their approach by estimating optimal parameter values for noise classification. Their approach achieved a noise classification accuracy of 85% to 100%.

Bountourakis et al. [22] aimed to develop methods for automatic recognition and classification of discrete environmental sounds, including those found in urban and rural audio scenes. Their study showed that the three algorithms used, namely ANN, k-NN, and SVM, performed well in terms of their recognition rate.

Sparke [23] aimed to investigate the capabilities of machine learning models in identifying industrial and environmental noise sources. The study's initial results indicate that these models are effective in the classification process, and their source contribution assessments are consistent with manually generated assessments.

Demir et al. [24] proposed a method for environmental sound classification using deep features extracted from data collected by a CNN model. The model was trained with images from the spectrogram, and the feature vector was computed by considering the connected layers of the model. To test the method, the feature set of the random subspaces of the KNN ensembles was taken into account. The experiments showed that the proposed model achieved a classification accuracy of 96.23% and 86.70%. The existing literature on noise pollution comprises a wide range of studies, but some of them have their drawbacks. For instance, expensive hardware sensors are commonly utilized, which can be unscalable and unsuitable for noise classification. In contrast, this project aims to utilize low-cost hardware to conduct noise classification and gather data [25].

Albaji et al. [26] have presented a machine learning approach to monitoring and classifying noise pollution. Both monitoring and classification methods have been implemented in MATLAB. The researchers have generated code to monitor all types of noise pollution from the collected data, and the machine learning algorithm was trained to classify these data. The ML algorithms showed promising performance in monitoring different sound classes such as highways, railways, trains, birds, airports, and more. The findings suggest that machine learning "ML" can be effectively utilized in monitoring and measuring noise pollution, and improvements can be made by enhancing the methods used to collect data. This could result in the development of more machine learning platforms to create a relevant environment with less noise pollution.

Ali et al. [27] used an ML approach evaluated with a dataset of only 4 sound samples grouped into four sound classes. They used Mel-frequency cepstral coefficients for feature extraction and supervised algorithms that are SVM, KNN, Bagging, and RF. However, their research has some limitations. Only four types of sounds were tested using four ML algorithms, and the results were found to be less accurate than initially reported after further investigation (73%, 89%, 91%, 90%). Additionally, the data used were not associated with IP addresses and linked to incorrect locations.

Mishra et al. [28] created an RVFL which is a model widely used for solving real-life regression and classification problems. Unfortunately, it is not able to reduce the effects of noisy data on the classification process. This paper presents a new IFRVFLC, which aims to improve the RVFL network's performance in binary classification. The training sample in IFRVFLC is composed of a fuzzy number with a membership degree and a nonmember degree. The distance from the training center is regarded as the membership degree of the pattern. On the other hand, the nonmember degree is determined by taking into account the total number of adjoining points. The performance of the IFRVFLC model against various support vector machines and kernels ridge regression was analyzed. The results of the study revealed that the proposed model is very user-friendly. It is also compared with other models such as the intuitionistic fuzzy SVM and the RVFL networks.

Hazarika and Gupta [29] proposed a novel KRR model, based on an affinity-based approach, to address the binary CIL problem. Their proposed model, AFKRR, considers the affinity of the majority class data points of the training samples and predicts its future performance. They evaluated the AFKRR using various metrics, such as the area under the curve (AUC), F-measure, and geometric mean, and compared its performance with other similar models, including support vector machine, affinity and class probability-based fuzzy SVM, KRR, intuitionistic fuzzy KRR, and CIL models. The results showed that AFKRR performs well in some real-world datasets when compared to the other models.

Hazarika and Gupta [30] highlight the challenge of handling feature noise in biomedical datasets when classifying them using machine learning models. They mention that the RVFL model is commonly used for classification and regression tasks, but its performance is negatively impacted by noisy data. To address this issue, the researchers propose a novel method to improve the RVFL's performance on noisy datasets.

Borah and Gupta [1] proposed an efficient classification method called ACFSVM, which is based on the affinity and class likelihood-based fuzzy vector machine. In their paper, they introduced two different class probability-based approaches to tackle the class imbalance problem. The first approach employs a cost-effective learning method, while the second approach uses a novel probability equation to adjust the class size. They reduced the sensitivity of various samples to noise and outliers by using the affinity of each sample to its class, which is obtained through the use of a support vector machine. The first approach is to use fuzzy membership values to transform the probability of class interactions into a standard LS-SVM-type formulation. It then introduces a new term to describe the effect of class imbalance on the performance of the system. The second approach reduces the outlier and noise sensitivity of the loss function of the first method by truncating it at a specified score. This method also handles the outliers and noise concerns at the optimization level. The second approach is to use a nonconvex structure as the basis for resolving the loss function. This method is supported by the concave-convex procedure. In order to evaluate the effectiveness of the two approaches, a number of simulations have been conducted on real-world and artificial datasets.

3. Proposed Machine Learning Based Approach for Noise Classification

In this section, five supervised classification methods that are commonly used in the classification of various types of objects are discussed. These include support vector machine, KNN, decision tree, logistic regression, and random.

(1) Support Vector Machines (SVM): Support vector machines are commonly used to solve classification problems. A supervised algorithm is used for performing such tasks. The goal of the SVM algorithm is to find the optimal hyperplane for training. It can then divide the data into two classes. There are many hyperplanes that can handle all the training data, but the one that leaves the most margin between itself and the nearest samples is the best choice [57]. The "support vector machine" (SVM) is a supervised machine learning technique that may be used for classification and regression tasks. It is, however, largely employed in categorization difficulties. Each data item is plotted as a point in n-dimensional space (where n is the number of features present), with the value of each feature being the value of a certain coordinate in the SVM algorithm. Then, classification is accomplished by locating the hyperplane that best distinguishes the two classes. The SVM algorithm's purpose is to find the optimum line or decision boundary for categorizing n-dimensional space so that fresh data points can be placed in the proper category in the future. A hyperplane is the optimal choice boundary.

(2) K-Nearest Neighbors (KNN): KNN is a relatively easy algorithm to use in machine learning. It takes into account the minimum distance between the training points and the test point and then produces a class mark for the test point depending on the class of the nearby k-nearest neighbors. The KNN is considered a lazy algorithm, which means that it does not take into account the training data points before it tests. This means that the training phase of the algorithm is very fast, and it can be used to reduce the training task to memorizing the training points. However, the testing phase of the algorithm is very expensive, and it requires a lot of time to perform. Furthermore, to store all of the training points, there is a need for more memory. KNN is an acronym for "K-nearest neighbors." It is a machine-learning algorithm that is supervised. The method can handle classification as well as regression problem statements. The sign "K" represents the number of nearest neighbors to a new unknown variable that must be predicted or categorized. The KNN algorithm calculates the distances between a query and all instances in the data and then selects the K number of examples closest to the query. It then proceeds to vote for the most frequent label (in classification) or average the labels (in regression). Being a nonparametric method, K-NN does not make any assumptions about the underlying data. Additionally, it is referred to as a lazy learner algorithm because it does not learn instantly from the training set. Instead, it stores the dataset and takes action during classification. During the training phase, KNN

algorithm saves the dataset and classifies it into a category that is similar to the incoming data. It is easy to create and comprehend, but it has the main disadvantage of being significantly slower as the amount of data in use increases.

(3) Decision trees (DT) are a supervised machine learning technique that trains models using labelled input and output datasets. The method is primarily used to solve classification issues, which involve using a model to categorize or classify an item. Decision trees are a form of predictive modeling that helps map the various options or solutions to a specific outcome. A decision tree is made up of different nodes, with the root node usually being the entire dataset in machine learning. Each internal node represents a criterion on a predictor. A leaf node, which contains the class label, represents the endpoint of a branch or the final outcome of a series of decisions. The decision tree does not branch further from a leaf node.

(4) Logistic regression is a common machine learning algorithm that falls within the supervised learning approach. It is used to forecast the categorical dependent variable from a group of independent factors. It forecasts a categorical dependent variable's outcome. As a result, the conclusion must be categorical or discrete. It can be Yes or No, 0 or 1, True or False, and so on, but instead of presenting the precise values like 0 and 1, it presents the probability values that fall between 0 and except for how they are employed, and logistic regression and linear regression are quite similar. Logistic regression is used to solve classification problems, while linear regression is used to solve regression problems. In logistic regression, instead of fitting a regression line, an "*S*" shaped logistic function is fitted, which predicts two maximum values (0 or 1) while generally predicting the likelihood of an outcome. The curve of the logistic function reflects the likelihood of the various outcomes, and the one with the highest probability is chosen as the output.

(5) A random forest is a supervised machine learning algorithm that employs decision tree algorithms. It is an approach for solving regression and classification problems that uses ensemble learning, a technique that combines several classifiers to solve complex problems. The random forest algorithm is made up of several decision trees. The "forest" created by the random forest algorithm is trained using bagging, a meta-algorithm that increases the accuracy of machine learning algorithms using an ensemble approach. The algorithm determines the outcome based on the predictions of the decision trees. It forecasts by averaging or averaging the output of multiple trees. The accuracy of the output improves as the number of trees increases. Random Forest (RF): The random forest algorithm is a type of decision tree that is related to the bag of decision trees. It can be used to analyze trees that are close to each other. The RF seeks to fix this issue by implementing a random forest that considers only a subset of the features that are randomly selected from the training subsets to determine the optimal split at each node in a tree. This method is different from the bag method, where all *M* features are selected to split the nodes in a tree. The random forest algorithm belongs to the ensemble methods family, and it is used to classify trees based on their predicted characteristics. However, because it uses many trees, it may take a long time to train. This is not a significant concern because it only requires one training session. Before training a machine learning model, it is important to format the data in a way that is simple to read and understand and to select the best machine learning algorithm for classifying noise. Figure 1 shows the proposed machine learning based approach for noise classification.

3.1. Environmental Noise Dataset. An extensive collection of information on the many kinds of noise recorded from numerous sites makes up the environmental dataset that was examined in this study. The dataset contains crucial geographic information, such as the city's name, longitude and latitude coordinates, and where the noise was captured. These specifics make it evident where the noise was captured and how it relates to other geographic places. Moreover, the dataset comprises noise-related features such as LA50 and LAeq, which aid in characterizing the noise and identifying its type and intensity. The research employed eighteen different categories of environmental sounds to provide clear and straightforward data. Each record in the dataset is labelled with the source of the noise, such as chatting, human children, footsteps, works, wind, and other sources, to facilitate easy comprehension. This dataset provides a well-balanced representation of various types of noise recorded from different locations, with almost equal numbers of recordings for each classification. In summary, this dataset is an invaluable resource for anyone interested in studying the distribution and characteristics of different types of noise captured from different locations. By conducting experiments with various environmental sounds, researchers can gain a better understanding of how this approach works. This is especially important as noises from these sources are prevalent in most Malaysian cities, as observed through the data analysis. Table 1 shows the classes of the sounds used.

3.2. Data Preparation. The Environmental Dataset used in this study came from https://Monitornoises.com, a website that collects information on the spatial distribution of property indices throughout Malaysian cities. All data produced by the https://MonitorNoises.com publisher are available on their website. The data are free and delivered under the ODbL terms. The dataset was initially in JSON format, but a Python program was used to convert it into a tabular form, which was then saved as a CSV file. The data of https://Monitornoises.com are gathered via a mobile application called NoiseCapture which is a free and open-

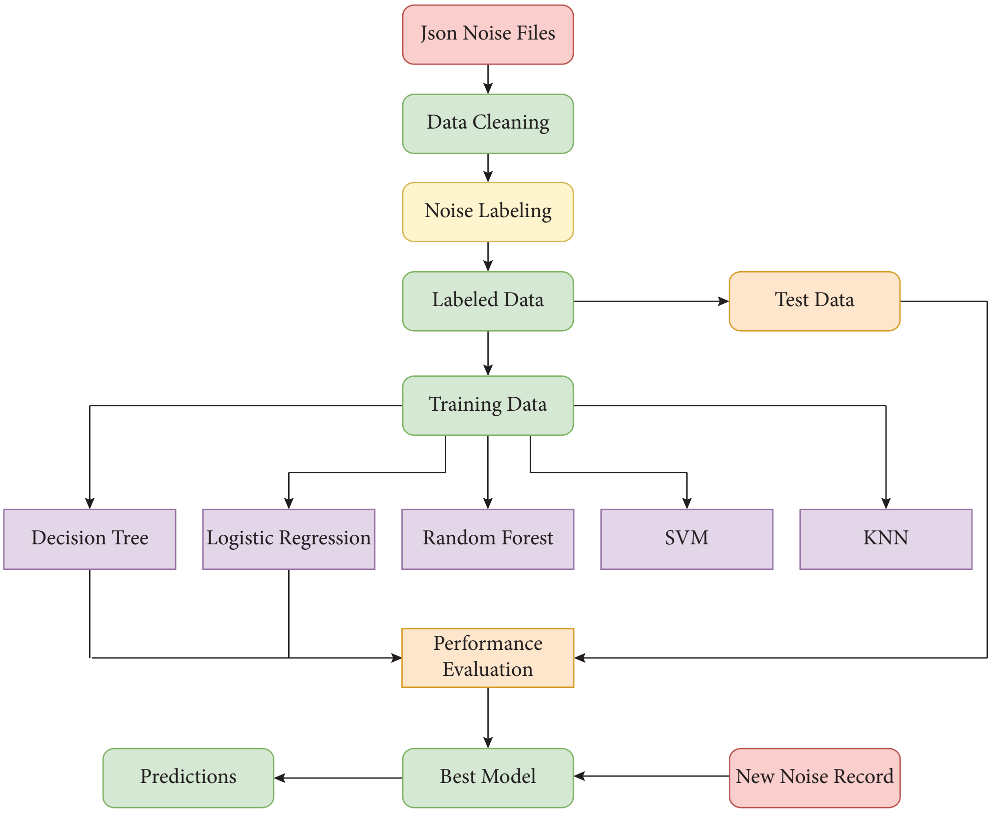


Figure 1: Proposed machine learning based approach for noise classification.

Table 1: Samples of sound.

No	Classes	Samples	Duration (min)
1	Johor_mall_Chatting	44	128
2	Kedah_beach_Wind	39	44
3	Kelantan_Police Headquarters_Works	9	28
4	Kuala Lumpur_Human Children	23	73
5	Labuan_wind	35	93
6	Melaka_Restaurant_Chatting	22	43
7	Negeri Sembilan_Works	13	78
8	Pahang_stade_Footsteps & Chatting	71	49
9	Perlis_ Industrial	45	78
10	Perak_Cars_Road	32	95
11	Pulau Pinang_hotel park_Test	56	61
12	Putrajaya_Works	98	102
13	Sarawak_Test	76	88
14	Selangor_ Industrial	54	87
15	Trengganu_Road	43	72
16	Sabah_airoport_Indoor	213	4322
Total	16 classes/16 cities	873	5441

source Android application that allows users to measure and share the noise environment. The data collected through the app are available in an open format. The researchers downloaded the dataset related to Malaysia as JeoJson Files and extracted the features related to each noise record using Python and the Pandas library. The resulting data were saved as an Excel file. Upon analysis, it was discovered that there were no missing values in the dataset, and null values did not need to be handled. Since all features were in numeric form, only the target variable was encoded, which was the label. The label encoding technique was used for the output column, which made it easier to analyze the data. The dataset is crucial since it provides insights into the distribution and characteristics of various noise types captured in different areas of Malaysian cities. The dataset is comprehensive and well-organized due to the geographic information, noise-related characteristics, and labelling of each record.

Table 2: Some of data captured.

<table>
<tr><td colspan="2">L80</td><td>L70</td><td>L60</td><td>L50</td><td colspan="2">L30 L40</td><td colspan="2">L10 L20</td><td colspan="3">Lmax Lmin L5</td><td colspan="2">Overall LEQ</td><td></td><td>Weighting</td></tr>
<tr><td>"</td><td>"</td><td>"</td><td>"</td><td>"</td><td colspan="2">"</td><td colspan="2">"</td><td>"</td><td colspan="2">"<i>t</i> (<i>s</i>)</td><td colspan="2">L90</td><td>"</td><td>"</td></tr>
<tr><td>"6.0</td><td>91.0"</td><td>"5.0</td><td>91.0"</td><td>"4.0</td><td colspan="2">"3.0 91.0"</td><td colspan="2">"2.0 91.0"</td><td>91.0"</td><td>"1.0</td><td colspan="2">91.0"</td><td>"0.0</td><td>(dB)"</td><td>Leq</td></tr>
<tr><td>"13.0</td><td>91.0"</td><td>"12.0</td><td>91.0"</td><td>"11.0</td><td colspan="2">"10.0 91.0"</td><td colspan="2">"9.0 91.0"</td><td>91.0"</td><td>"8.0</td><td colspan="2">91.0"</td><td>"7.0</td><td>"</td><td>91</td></tr>
<tr><td>"20.0</td><td>91.0"</td><td>"19.0</td><td>91.0"</td><td>"18.0</td><td colspan="2">"17.0 91.0"</td><td colspan="2">"16.0 91.0"</td><td>91.0"</td><td>"15.0</td><td colspan="2">91.0"</td><td>"14.0</td><td>"</td><td>91</td></tr>
<tr><td>"27.0</td><td>91.0"</td><td>"26.0</td><td>91.0"</td><td>"25.0</td><td colspan="2">"24.0 91.0"</td><td colspan="2">"23.0 91.0"</td><td>91.0"</td><td>"22.0</td><td colspan="2">91.0"</td><td>"21.0</td><td>"</td><td>91</td></tr>
<tr><td>"</td><td>"</td><td>"</td><td>"</td><td>"</td><td colspan="2">"</td><td colspan="2">"</td><td>"</td><td>"</td><td colspan="2">91.0"</td><td>"28.0</td><td>"</td><td>91</td></tr>
<tr><td></td><td>91 dB</td><td>91 dB</td><td>91 dB</td><td>91 dB</td><td colspan="2">91 dB 91 dB</td><td colspan="2">91 dB 91 dB</td><td>91 dB</td><td>91 dB</td><td colspan="2">91 dB</td><td>91 dB</td><td>"</td><td>91 dB</td></tr>
<tr><td colspan="2">L80</td><td>L70</td><td>L60</td><td>L50</td><td colspan="2">L30 L40</td><td colspan="2">L10 L20</td><td colspan="5">Overall LEQ Lmax Lmin L5 L90 "<i>t</i> (<i>s</i>)</td><td>"</td><td>Weighting</td></tr>
<tr><td>60.6"</td><td>"6.0</td><td>60.6"</td><td>"5.0</td><td>60.3"</td><td>"4.0</td><td>60.5"</td><td>"3.0</td><td>60.8"</td><td colspan="7">Leq (dB)" "0.0 60.3" "1.0 60.2" "2.0</td></tr>
<tr><td>62.0"</td><td>"13.0</td><td>61.2"</td><td>"12.0</td><td>61.2"</td><td>"11.0</td><td>60.8"</td><td>"10.0</td><td>60.6"</td><td colspan="7">"7.0 60.9" "8.0 60.9" "9.0</td></tr>
<tr><td>74.1"</td><td>"20.0</td><td>73.0"</td><td>"19.0</td><td>72.6"</td><td>"18.0</td><td>71.0"</td><td>"17.0</td><td>69.0"</td><td colspan="7">"14.0 61.5" "15.0 67.8" "16.0</td></tr>
<tr><td>69.7"</td><td>"27.0</td><td>69.6"</td><td>"26.0</td><td>69.3"</td><td>"25.0</td><td>67.3"</td><td>"24.0</td><td>66.4"</td><td colspan="7">"21.0 74.2" "22.0 69.4" "23.0</td></tr>
<tr><td>69.4"</td><td>"34.0</td><td>69.7"</td><td>"33.0</td><td>69.6"</td><td>"32.0</td><td>69.8"</td><td>"31.0</td><td>69.7"</td><td colspan="7">"28.0 69.6" "29.0 69.5" "30.0</td></tr>
<tr><td>70.0"</td><td>"41.0</td><td>69.6"</td><td>"40.0</td><td>69.3"</td><td>"39.0</td><td>69.1"</td><td>"38.0</td><td>69.5"</td><td colspan="7">"35.0 69.5" "36.0 69.4" "37.0</td></tr>
<tr><td>69.8"</td><td>"48.0</td><td>69.3"</td><td>"47.0</td><td>69.2"</td><td>"46.0</td><td>69.1"</td><td>"45.0</td><td>69.0"</td><td colspan="7">"42.0 69.6" "43.0 69.8" "44.0</td></tr>
<tr><td>70.3"</td><td>"55.0</td><td>69.8"</td><td>"54.0</td><td>69.8"</td><td>"53.0</td><td>69.9"</td><td>"52.0</td><td>69.8"</td><td colspan="7">"49.0 69.6" "50.0 70.3" "51.0</td></tr>
<tr><td>69.6"</td><td>"62.0</td><td>70.3"</td><td>"61.0</td><td>70.4"</td><td>"60.0</td><td>69.9"</td><td>"59.0</td><td>70.0"</td><td colspan="7">"56.0 70.1" "57.0 70.5" "58.0</td></tr>
<tr><td>80.0"</td><td>"69.0</td><td>69.7"</td><td>"68.0</td><td>70.0"</td><td>"67.0</td><td>69.7"</td><td>"66.0</td><td>69.6"</td><td colspan="7">"63.0 70.4" "64.0 69.8" "65.0</td></tr>
<tr><td>78.3"</td><td>"76.0</td><td>78.3"</td><td>"75.0</td><td>78.1"</td><td>"74.0</td><td>78.5"</td><td>"73.0</td><td>78.0"</td><td colspan="7">"70.0 77.2" "71.0 76.9" "72.0</td></tr>
<tr><td>78.1"</td><td>"83.0</td><td>78.0"</td><td>"82.0</td><td>77.8"</td><td>"81.0</td><td>78.0"</td><td>"80.0</td><td>77.8"</td><td colspan="7">"77.0 79.5" "78.0 79.2" "79.0</td></tr>
<tr><td>76.7"</td><td>"90.0</td><td>76.7"</td><td>"89.0</td><td>76.5"</td><td>"88.0</td><td>76.6"</td><td>"87.0</td><td>76.8"</td><td colspan="7">"84.0 76.7" "85.0 77.7" "86.0</td></tr>
<tr><td>76.9"</td><td>"97.0</td><td>76.7"</td><td>"96.0</td><td>77.1"</td><td>"95.0</td><td>77.3"</td><td>"94.0</td><td>77.0"</td><td colspan="7">"91.0 76.8" "92.0 76.9" "93.0</td></tr>
<tr><td>77.0"</td><td>"104.0</td><td>77.1"</td><td>"103.0</td><td>76.9"</td><td>"102.0</td><td>77.1"</td><td>"101.0</td><td>77.1"</td><td colspan="7">"98.0 76.9" "99.0 77.0" "100.0</td></tr>
<tr><td>70.4"</td><td>"111.0</td><td>68.8"</td><td>"110.0</td><td>67.3"</td><td>"109.0</td><td>69.4"</td><td>"108.0</td><td>75.2"</td><td colspan="7">"105.0 77.2" "106.0 77.9" "107.0</td></tr>
<tr><td>88.2"</td><td>"118.0</td><td>89.5"</td><td>"117.0</td><td>86.0"</td><td>"116.0</td><td>81.5"</td><td>"115.0</td><td>80.0"</td><td colspan="7">"112.0 69.1" "113.0 73.4" "114.0</td></tr>
<tr><td>"</td><td>72.8"</td><td>"124.0</td><td>76.4"</td><td>"123.0</td><td>76.0"</td><td>"122.0</td><td>76.8"</td><td>"121.0</td><td colspan="7">"119.0 83.3" "120.0 74.9"</td></tr>
<tr><td colspan="3">68.9 dB 61.6 dB</td><td colspan="2">69.8 69.5 dB</td><td>70.3 dB</td><td>73.6 dB</td><td>76.7 dB</td><td>77.2 dB</td><td colspan="7">76.4 dB 90 dB 58.8 dB 80 dB 78.2 dB</td></tr>
</table>

3.3. Data Analysis. The 209 records and 13 columns in the dataset examined in this study contained diverse data regarding the noise levels observed at various places throughout Malaysia. The dataset contains geographic information on the locations where the noise recordings were made, such as the city's name, longitude, and latitude coordinates. The dataset contained various noise-related features such as "LA50," a statistical descriptor that represents the sound level exceeded for 50% of the measurement period, and "LAeq," which measures the constant noise level that would generate the same total sound energy over a specified period. The records in the dataset were classified and assigned a label identifying the source of the noise, such as conversation, wind, or other noise sources. This labelling of the different types of noise captured in the sample makes it easy to categorize and analyze the data.

Before proceeding with any machine learning model, it is essential to perform some critical data analysis steps to understand the data better. In this case, it is evident that chatting noise is the most frequent, while children's noise is less common. Furthermore, the number of noises captured for each of Malaysia's cities was counted, as shown in Table 2, and it is apparent that Sabah and Sarawak have captured more sounds.

4. Results

4.1. Band Levels. An ensemble model is a type of learning algorithm that uses multiple learning methods to improve its predictive performance. It can also be useful in improving accuracy in certain cases. Figures 2 and 3 illustrate the band levels of some of the noise samples captured over Malaysian cities. The band levels represent the output column and the frequency of each class occurring in the output column. The pie chart in Figure 2 displays the frequency distribution of each noise type. The highest frequency is the chatting noise type, followed by the indoor noise type. The dataset had 209 entries categorized into chatting, human children, footsteps, works, wind, industrial, test, road, indoor, motorbike, lawnmower, motorway, bar, aircraft, alarm, gun, call wave, and garbage truck. The pie chart illustrates that each label had about an equal proportion of records, with no particular noise source dominating over the others. Upon closer inspection of the figure, certain labels have a slightly lower share than others. For example, call wave has the highest proportion (6.898%), while footsteps have the lowest proportion (4.437%). Overall, the pie chart reveals that the labels are distributed rather evenly, indicating that each noise source was captured approximately the same number of times in the dataset.

Figure 3 depicts a pie chart presenting the percentage of recordings used in the noise level experiment collected from different Malaysian cities. The chart provides an overview of how the noise levels were distributed among various cities in Malaysia. As per the data, the majority of the samples were obtained from Sarawak, Sabah, and Perak, whereas the least number of recordings were collected from Labuan, Trengganu, Putrajaya, Kuala Lumpur, and Negeri Sembilan. This implies that the noise level in these cities was either not as

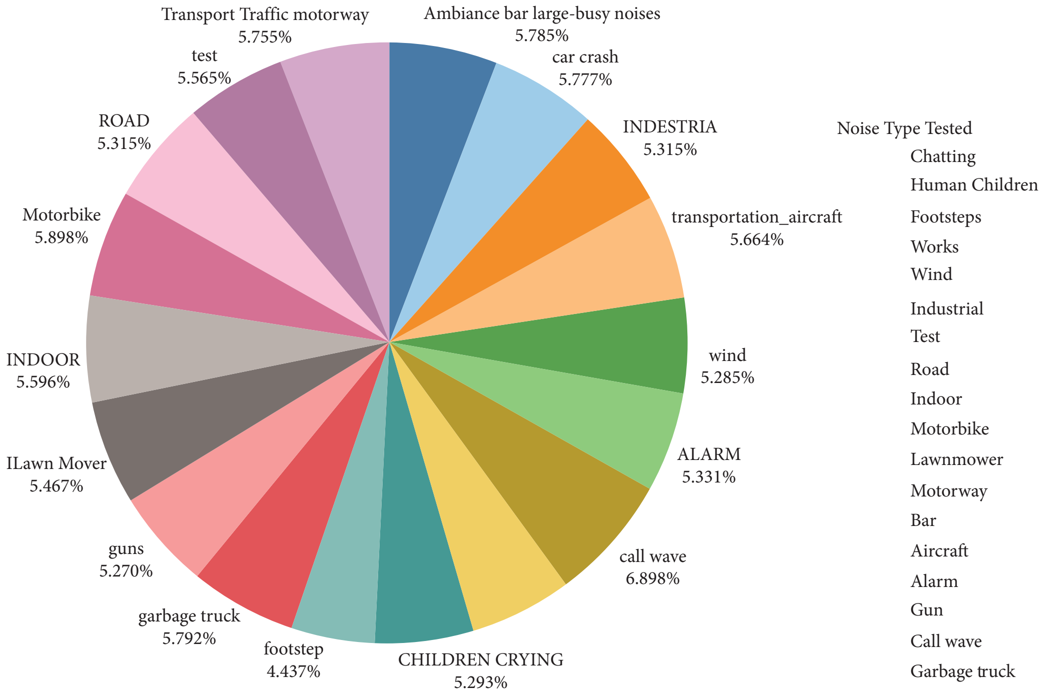


Figure 2: Band levels.

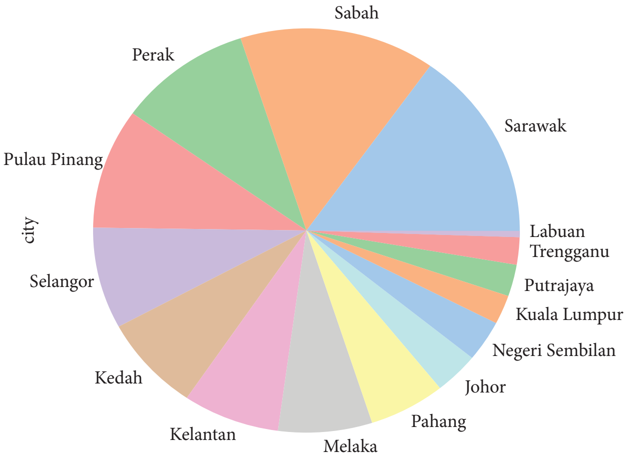


Figure 3: Number of noises over Malaysian cities.

significant or not as frequently recorded as in other places. It is worth noting that the recordings were taken from a total of 16 distinct Malaysian cities, indicating the comprehensiveness of the study. Figure 3's pie chart offers important insights into the distribution of noise levels across the nation's cities and aids in determining which cities have higher levels of noise pollution.

4.2. Points. The "Point" feature of the investigated dataset was analyzed separately from other features. Figure 4(a) displays the distribution of the "Point" feature, indicating that it can range from a minimum value of 0 to a maximum value of 4. The distribution of "Point" values is multimodal, with one peak between 0 and 1 and the majority of the values concentrated in the range between 3 and 4. Figure 4(b) illustrates the distribution of the "Point" feature for each of the 16 class label categories. The distribution for the classes of humans and children was concentrated at a single point, which was 0. In contrast, the "Point" value was relatively consistent for all other classes. Figure 4(c) displays the average value of the "Point" feature for each class. With the exception of toddlers and humans, all classes had an average between 1.5 and 2.0. Both toddlers and humans had an average value of 0 for the "Point" feature. Meanwhile, Figure 4(d) shows the median value and five-point summary of the "Point" feature for each class. The median value of the "Point" feature for all classes, except for toddlers and humans, was between 1.5 and 2.0. The median value of the "Point" feature was observed to be 0 for both toddlers and

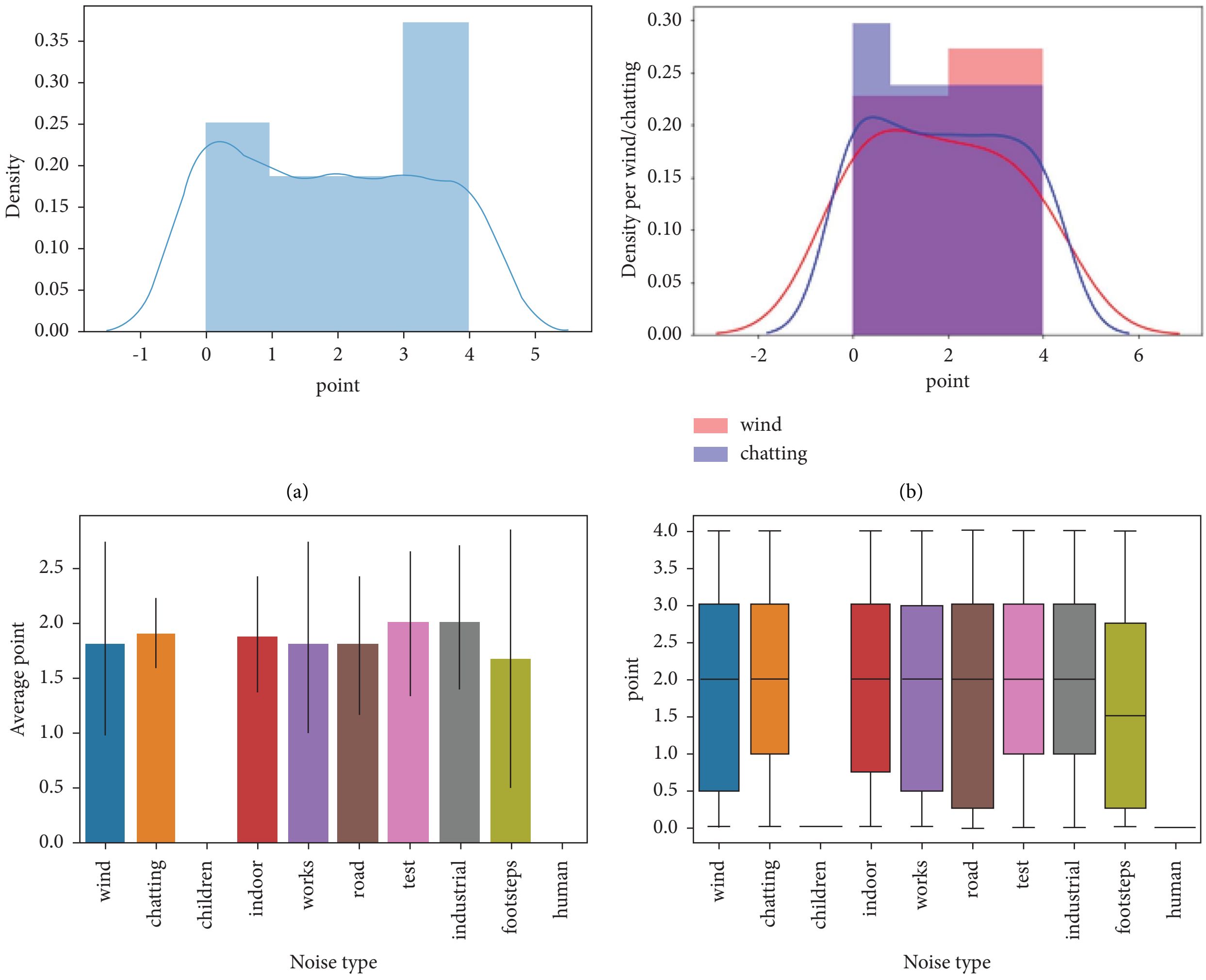


Figure 4: Plot for point: (a) point, (b) point, (c) noise type, and (d) noise type.

humans. Therefore, the study of the "Point" feature suggests that the value of "Point" for adults and children in the sample consistently had a value of zero.

4.3. *Properties_q.*

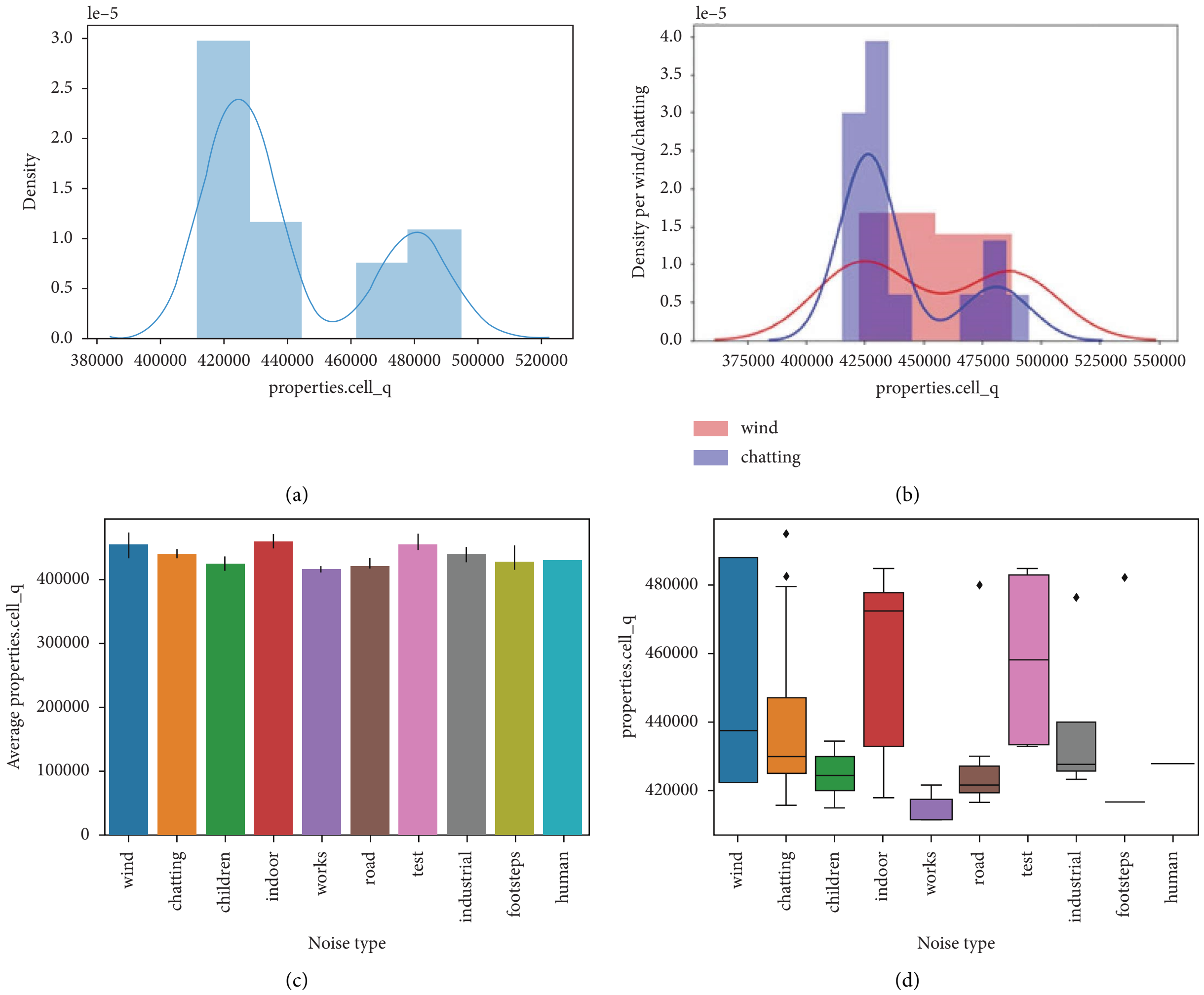
Figure 5(a) shows the distribution of "Properties q" values, ranging from 400000 to 500000. The distribution appears to be multimodal, with a larger peak around 420000 and a smaller peak around 490000. Additionally, there is a gap in the distribution between 450000 and 460000, indicating a lack of records for "Properties q" values in that range. Figure 5(b) displays the distribution of "Properties q" for the class labels "wind" and "chatting." The distribution for "wind" appears to be rather evenly, while for "chatting," it is substantially concentrated around 430000. Figure 5(c) presents the average value of "Properties q" for each class label, indicating that all classes have an average value above 400000. Finally, Figure 5(d) shows the median value and five-point summary of "Properties q" for various class labels. Although the median values are distributed among the different classes, some outliers can be observed for certain class labels, such as conversation, road, industrial, and footfall.

4.4. *Properties.Cell_r.*

Figure 6 displays the plots for Properties. cell_r. It also uses bivariate analysis with properties cell_q and other variables. The dataset's feature "cell r" was examined separately from other features. The distribution of "cell q" is shown in Figure 6(a) and has values ranging from 5000 to 35000, with a uniform distribution. Figure 6(b) shows the difference in distribution between the classes "wind" and "conversing," where the wind distribution is centered around 30000, while the talking distribution is uniform. Figure 6(c) shows the average value of "cell q" for each class, falling between 15000 and 30000. The median value and five-point summary of "cell r" for various classes are shown in Figure 6(d), indicating that the medians are dispersed across a range of values. The feature values are concentrated for the classifications "wind," "footfall," and "human."

4.5. *Properties.la50.*

In Figure 7(a), the feature "la50" is examined and shown as a uniform distribution of values between 40 and 110. The comparison of the distributions of wind and chatting in Figure 7(b) reveals that whereas the distribution of wind is uniform, the distribution of chatting



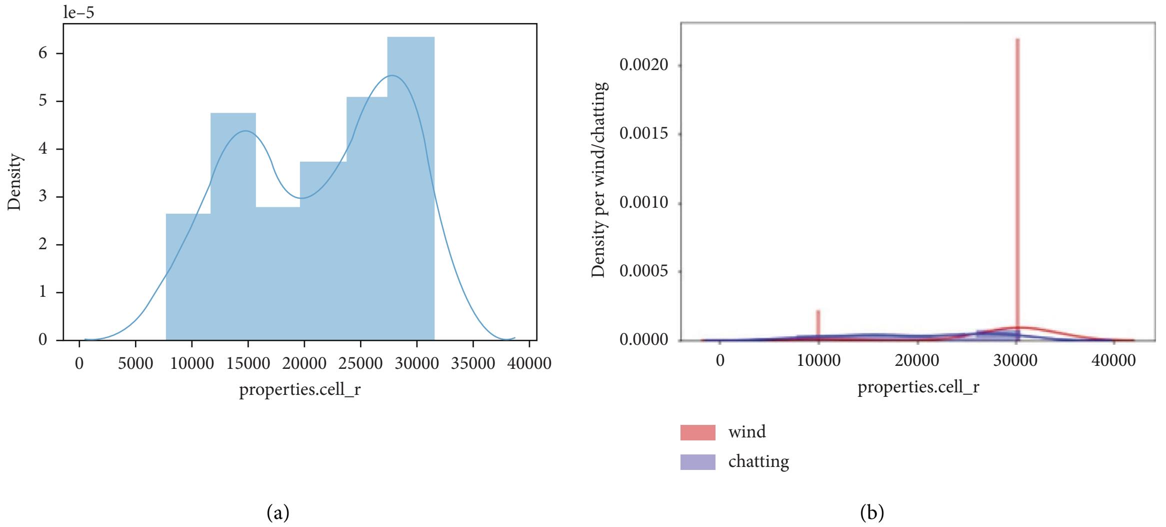
(a)

(b)

(c)

(d)

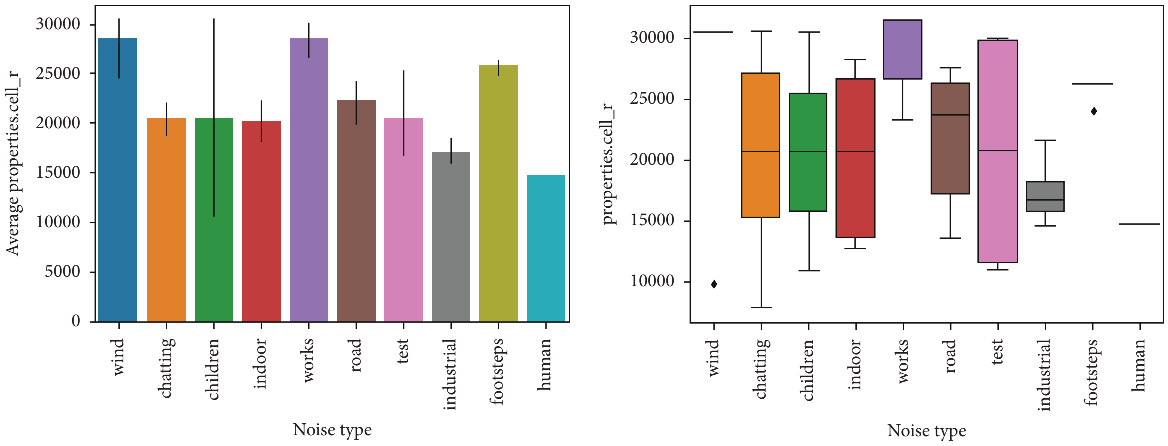
Figure 5: Plots for properties_q. (a) Properties.cell_q. (b) Properties.cell_q. (c) Noise type. (d) Noise type.



(a)

(b)

Figure 6: Continued.



(c) (d)

Figure 6: Plots for properties.cell_r. (a) Properties.cell_r. (b) Properties.cell_r. (c) Noise type. (d) Noise type.

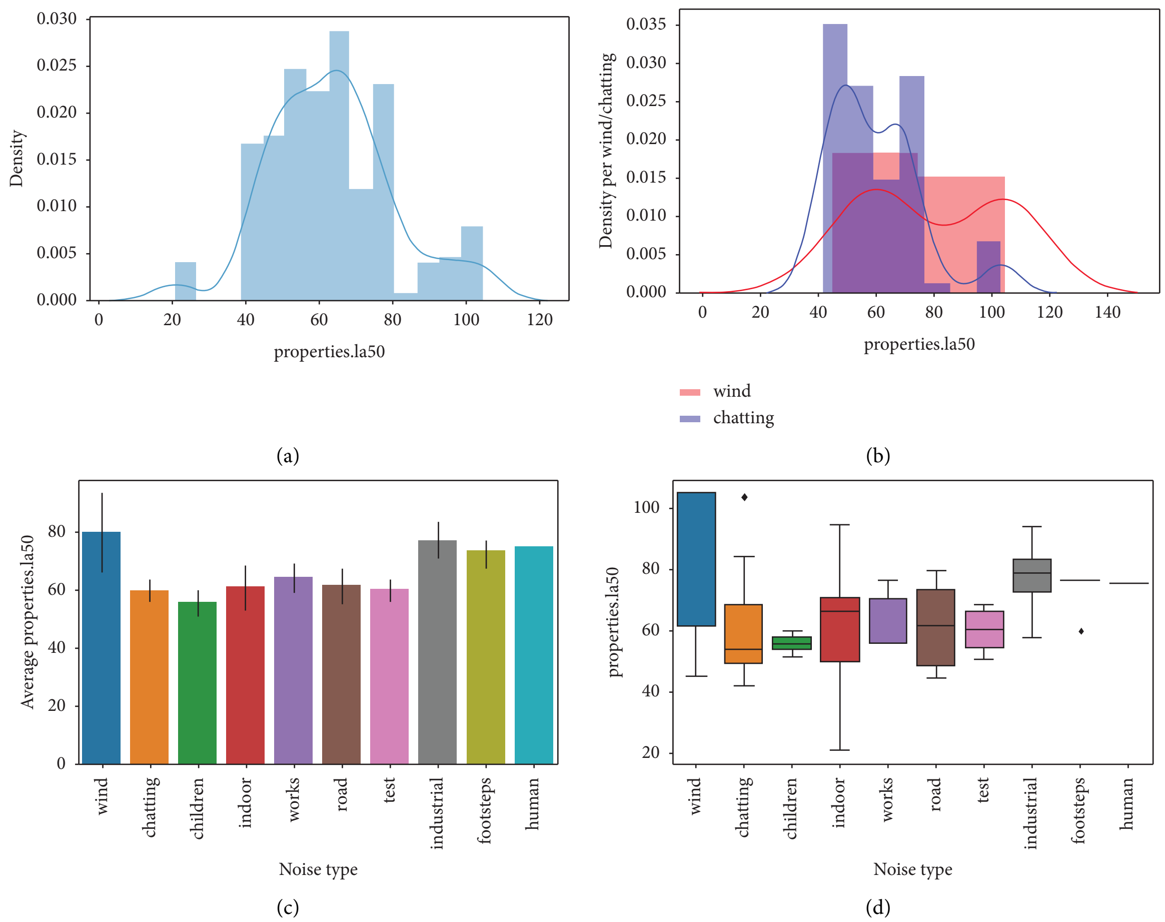
is concentrated at two places between 50 and 70. The average value of "la50" for several classes is shown in Figure 7(c), with average values between 50 and 80. The median value and five-point summary of "la50" for the various classes in Figure 7(d) show that the medians are dispersed across the classes with varying values and concentrated for classes such as children, footsteps, and humans. Additionally, outlier values are seen when talking and walking.

4.6. Properties.Laeq. The distribution of the feature "laeq" can be observed in Figure 8(a), with most values ranging between 40 and 110. The distribution of this feature is uniform, with a median of around 70. The difference in distribution between wind and chatting is depicted in Figure 8(b), with chatting having a concentrated distribution at two spots, the first one being around 55 and the second one being around 70. The average value of "laeq" for the various classes is depicted in Figure 8(c) as ranging between 50 and 80. The median value and five-point summary of "laeq" for several classes are given in Figure 8(d), with medians showing a range of values and a concentration of values for the human, kid, and footstep classes. In the cases of talking, taking a test, and taking steps, outlier results are seen. This feature's behavior is discovered to be extremely similar to the "la50" feature.

4.7. Properties.Iden. Figure 9 displays the plots for Properties-iden and uses bivariate analysis with Properties-Iden and other variables. The first plot shows the data distribution of "Properties. iden" via a histogram. Since it consists of numeric variables, a histogram is applied to display the frequency distribution. All values are concentrated at a single point of 0.0, and there are no other values. The second graph shows the density of points per wind/chatting which is the same for each value of "Properties. iden" and hence follows a rectangular curve. The third graph shows the frequency distribution of each noise type with respect to Properties. iden in the form of a barplot. The fourth graph shows the boxplot of each noise type with Properties. iden. The boxplot shows that each noise type contains only the value zero. All in all, it can be concluded that the values are very constant in "Properties. iden" and not significant in the analysis.

4.8. Properties.Measure_Count. The distribution of the feature "measure count" is shown in Figure 10(a). According to the histogram, most of the values for this feature fall between 0 and 1500, with a notable peak at around 0. The histogram shows another bar around the number 5500. The difference in distribution between wind and conversation is shown in Figure 10(b). The distribution of wind appears to be bimodal, with peaks at 0 and 1000. The distribution of talking, on the other hand, is characterized by a long peak at around 0 and shorter peaks between about 0 and 2000 as well as at about 5000. The average value of "measure count" for different classes is shown in Figure 10(c), with the majority of the classes having averages between 0 and 100. The averages are slightly higher, at about 300, 450, and 300, respectively, for wind, conversation, and children. Finally, Figure 10(d) displays the median value and five-point summary of "measure count" for several classes. The median values for most classes are around 0, but there are some anomalous values in conversation.

4.9. Properties.First_Measure_Epoch. The distribution of the characteristic "first measure epoch" is shown in Figure 11. The feature has a bimodal distribution, as shown in part (a) of the figure, with peaks around 1.55 and 1.65, and minimum and maximum values of 1.5 and 1.65, respectively. In part (b), the distributions of wind and chatting are compared. Both distributions display a pattern that is somewhat bimodal for the wind and singularly modal for the chatting. The average value of the feature for each class is shown in part (c), and it is discovered that the averages range between 1.5 and 1.6. Part (d) displays the median value and five-point description of the feature for several classes, with certain classes, such footsteps and humans, having more condensed distributions than others. Additionally, there are few outliers in the distribution of the feature for footsteps.



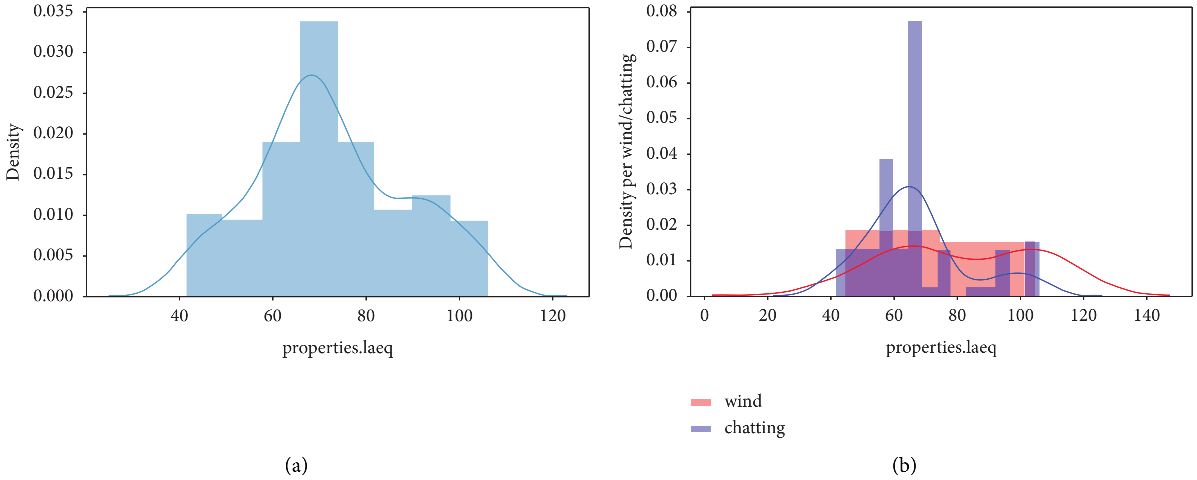
(a)

(b)

(c)

(d)

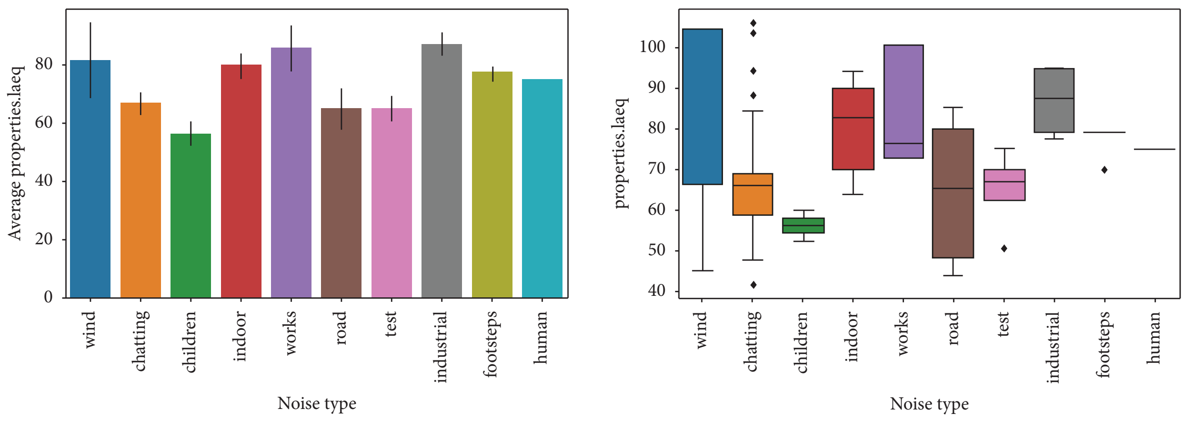
Figure 7: Plots for properties.la50. (a) Properties.la50. (b) Properties.la50. (c) Noise type. (d) Noise type.



(a)

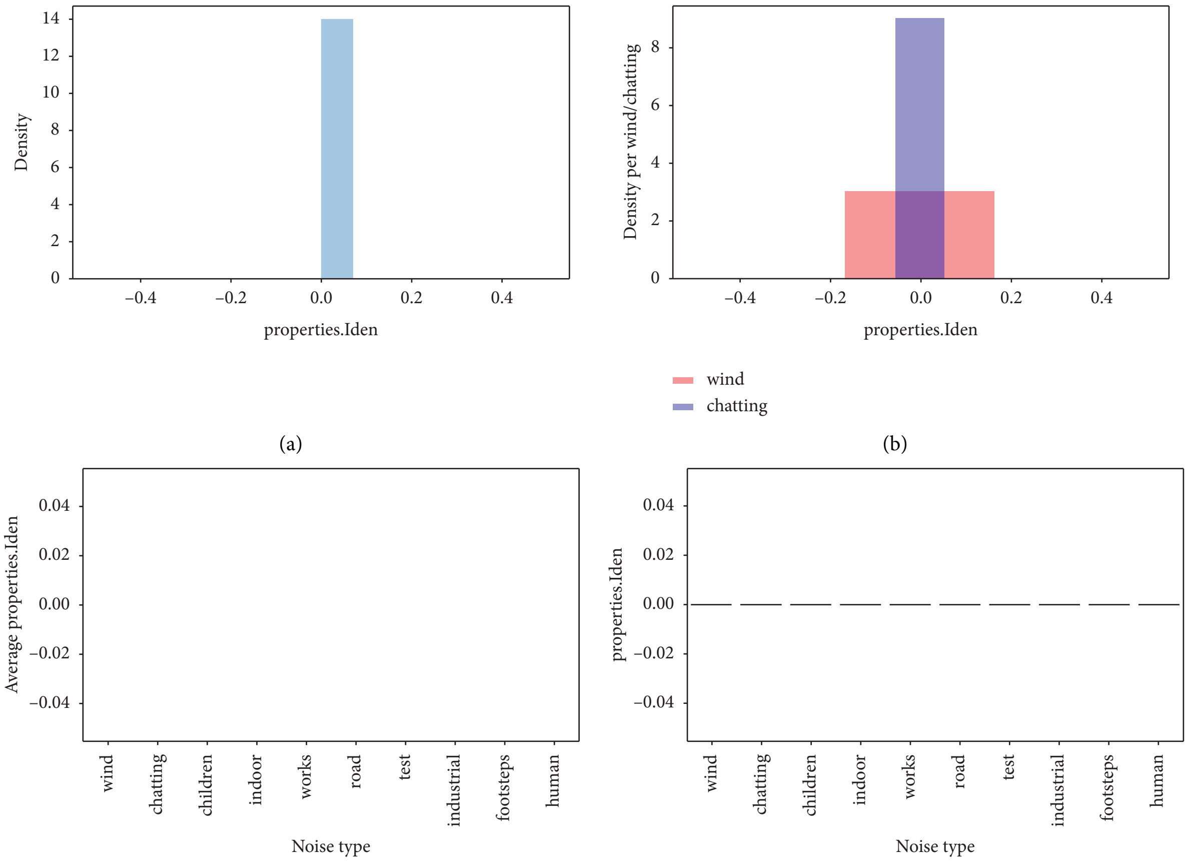
(b)

Figure 8: Continued.



(c) (d)

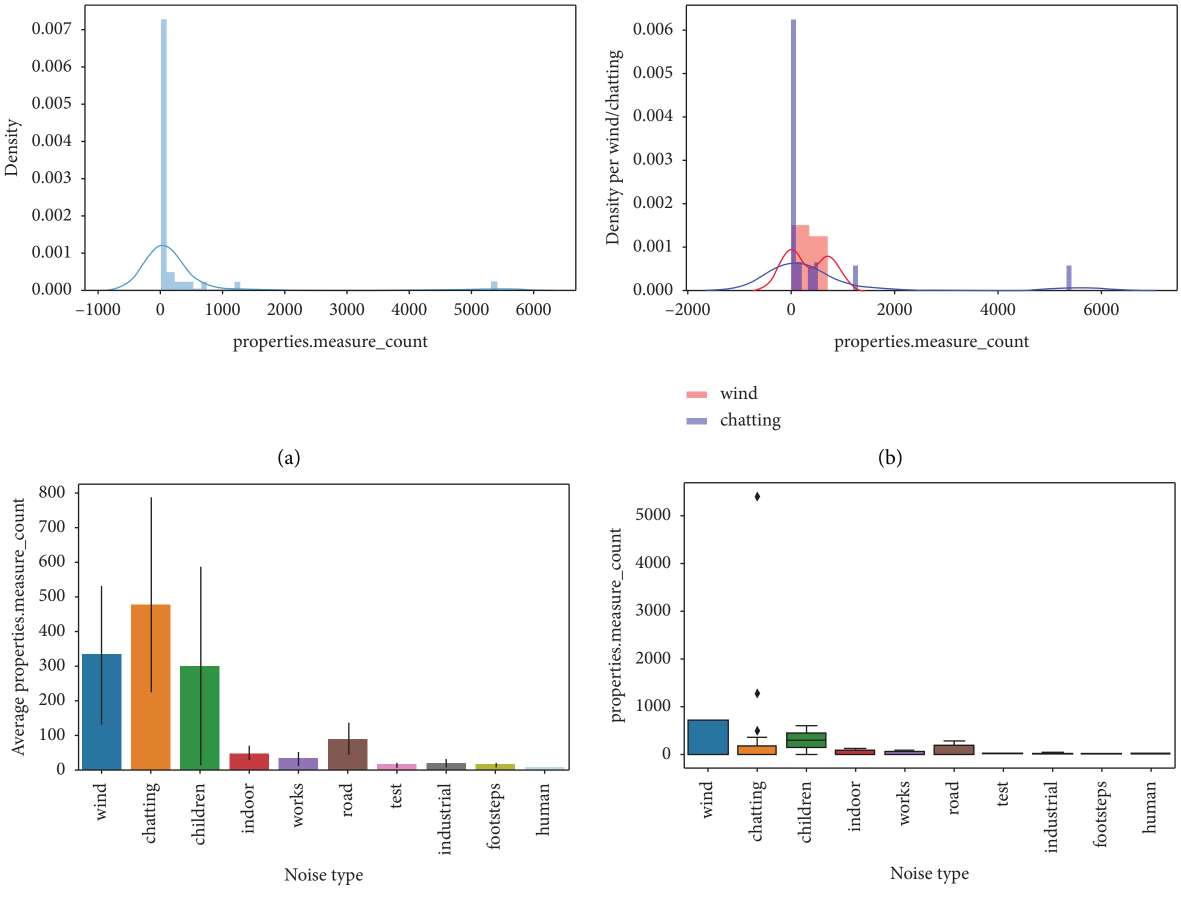
Figure 8: Plots for properties.laeq. (a) Properties.laeq. (b) Properties.laeq. (c) Noise type. (d) Noise type.



(c) (d)

Figure 9: Plots for properties.iden. (a) Properties.iden. (b) Properties.iden. (c) Noise type. (d) Noise type.

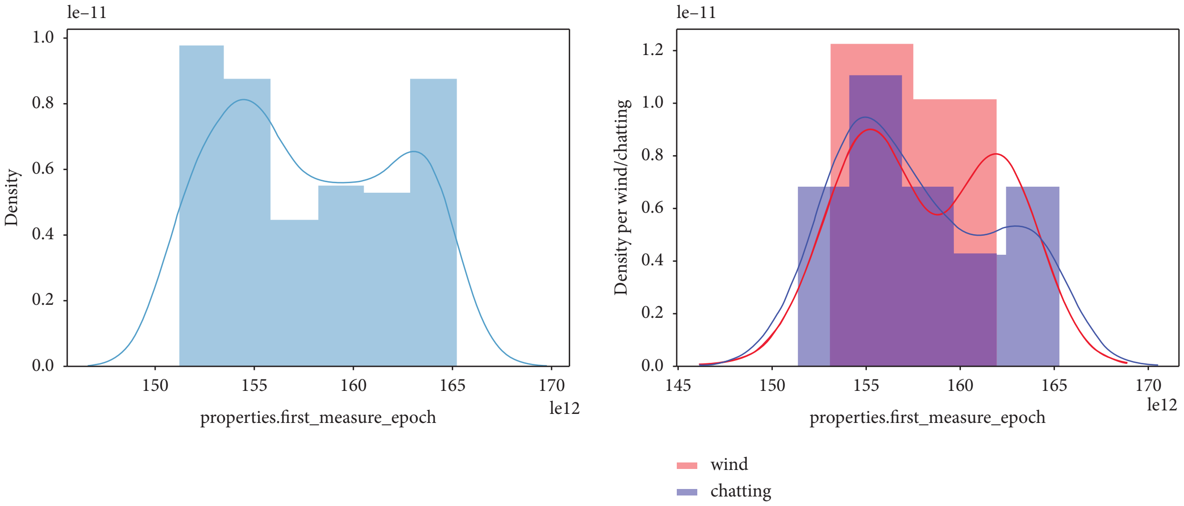
4.10. Properties.Last_Measure_Epoch. The distribution of the feature "last measure epoch" is examined in Figure 12(a). The range of values for this characteristic is determined to be 1.5 to 1.65, with two peaks forming a bimodal distribution at 1.55 and 1.65. Similarities may be seen between the distributions of the wind and chatting classes, with the wind class exhibiting a somewhat bimodal pattern as opposed to the chatting class's singular modal distribution as shown in Figure 12(b). The average value of the "last measure epoch" for each class is illustrated in Figure 12(c), and it can be seen that the average value for each class ranges between 1.5 and 1.6. The median value and five-point summary of the "last measure epoch" for various classes are shown in Figure 12(d), indicating that the median value can differ significantly depending on the class, with some classes having a more concentrated distribution than others. The



(a) (b)

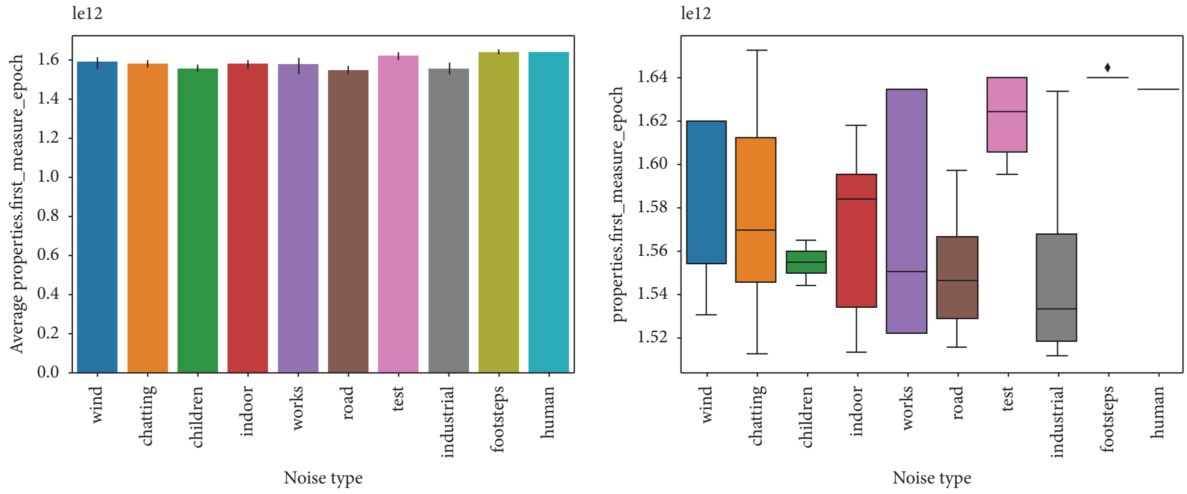
(c) (d)

Figure 10: Plots for properties.measure_count. (a) Properties.measure_count. (b) Properties.measure_count. (c) Noise type. (d) Noise type.



(a) (b)

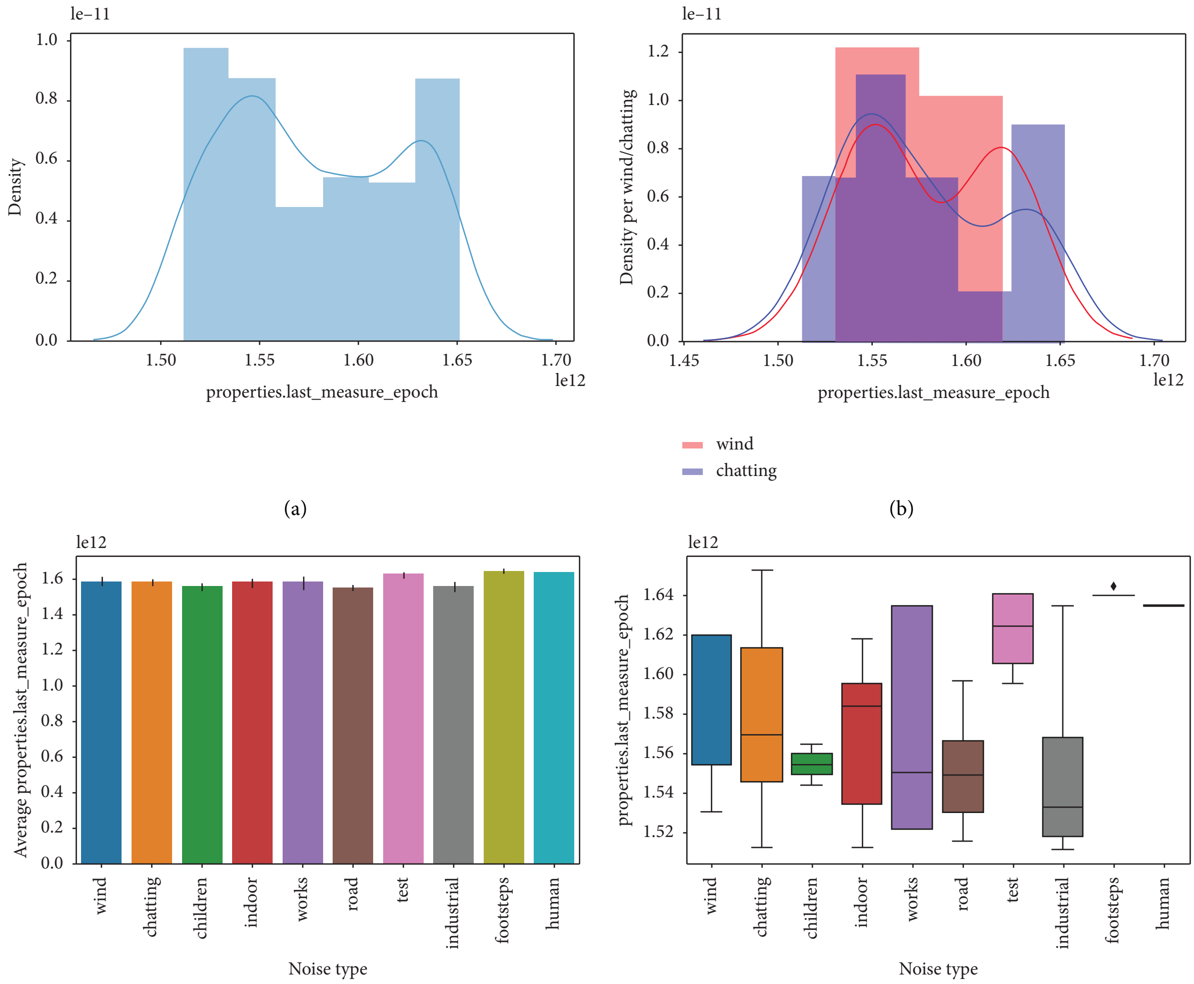
Figure 11: Continued.



(c)

(d)

Figure 11: Plots for properties.first_measure_epoch. (a) Properties.first_measure_epoch. (b) Properties.first_measure_epoch. (c) Noise type. (d) Noise type.



(a)

(b)

(c)

(d)

Figure 12: Plots for properties.last_measure_epoch. (a) Properties.last_measure_epoch. (b) Properties.last_measure_epoch. (c) Noise type. (d) Noise type.

distribution of the persons and footprints classes is quite concentrated, and there are some outliers in the footprints class as well. The feature "first measure epoch" is observed to exhibit behavior that is similar to this behavior, and both features appear to do so.

4.11. Latitude. The distribution of the feature "latitude" is shown in Figure 13(a). The graphic illustrates the feature's minimum and maximum values, which are 1.5 and 6.5, respectively. With two peaks situated at 3 and 6, it appears to be a bimodal distribution. The difference in distribution between wind and conversation is shown in Figure 13(b). While the distribution of talking is homogeneous, that of wind has a peak and concentration at about 6. The average value of "latitude" for each class is shown in Figure 13(c). The range of averages across all classes is between 3 and 6. The median value and five-point summary of "latitude" for various classes are shown in Figure 13(d). With wind, footsteps, and humans having highly concentrated distributions and several outliers as well, it can be seen that the median value of the features differs significantly among different classes.

4.12. Longitude. The distribution of the feature "longitude" is shown in Figure 14(a). The figure displays a bimodal distribution with a larger peak around 100 and a smaller peak around 115. Additionally, the feature has a minimum value of 100 and a maximum value of less than 120, as shown in Figure 14(b). The distribution of wind and chatting is compared in the same figure. It is observed that chatting has a peak and concentration at around 100 while the distribution of wind is uniform. Figure 14(c) displays the average value of "longitude" for each class, and it is clear that the average value for every class is higher than 100. Figure 14(d) demonstrates the median value and five-point summary of "longitude" for several classes. As can be observed, the distributions of footsteps and people are quite concentrated, while the median value of the feature differs substantially between classes. There are also outliers in certain classifications, including road, industrial, and footfall.

After performing exploratory data analysis, the correlation heatmap was examined, and various classification models were used to predict the types of noises.

5. Classification Models

Figure 15 illustrates the correlation matrix, which depicts the degree to which each predictor variable is correlated with the others. The primary objective of this project is to predict the "Label," which serves as the target column. To examine how the features are related to the Label, it has been encoded into a numeric column, and its correlation with other predictors is shown in Figure 16. Correlation indicates the extent to which changes in one variable are associated with changes in another variable. A correlation matrix is a table that displays the correlation coefficients between multiple variables. Each cell in the table represents the correlation between two variables. A correlation coefficient is a value between −1 and 1 that measures the strength and direction of a linear relationship between two variables. A coefficient of 1 implies a perfect positive correlation, a coefficient of −1 implies a perfect negative correlation, and a coefficient of 0 implies no correlation. Simply put, a correlation matrix is a tool that aids in understanding the relationship between various variables. In a correlation matrix, the variables are listed on both the x-axis and y-axis. Each variable appears only once on the x-axis and once on the y-axis. The values in the table cells, also known as correlation coefficients, correspond to the relationship between the two variables listed on the corresponding row and column. For instance, if the correlation matrix contains the variables A, B, and C, the top left cell of the matrix would show the correlation between variable A and variable B, and the cell below it would display the correlation between variable A and variable B, and so on. The diagonal of the matrix, from the top left to the bottom right, would be 1 because the correlation between a variable and itself is always 1. Figure 16 shows a correlation matrix where the diagonal values are all one. This is because the features are compared to themselves and have a perfect linear relation with themselves. The red values indicate a negative correlation, meaning that an increase in one feature results in a decrease in the other. Conversely, the green values indicate a positive correlation. The scale on the right side, ranging from −0.2 to 0.8, shows the gradient of colors used to understand the nature of the correlation between features. Dark colors represent high negative correlations, light colors represent high positive correlations, and medium-range colors indicate low or no correlation.

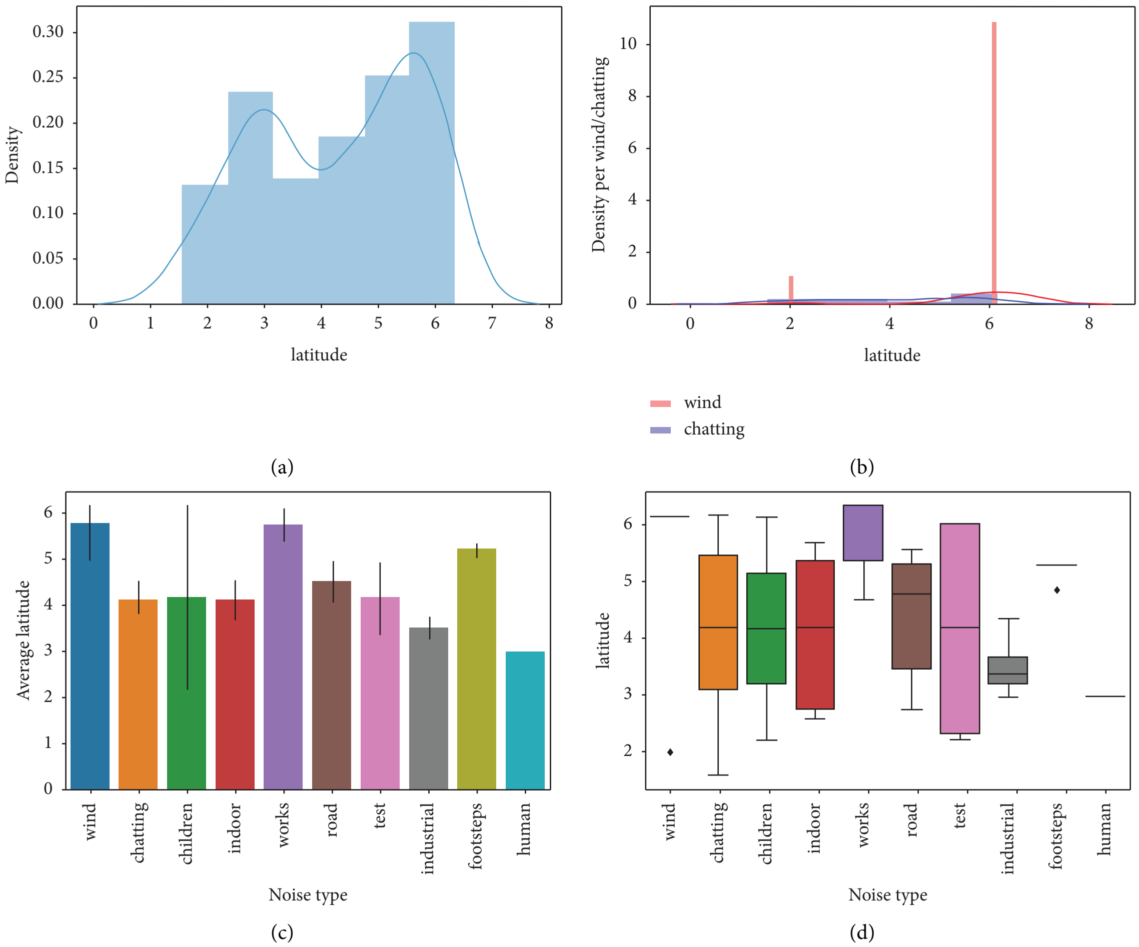
5.1. For Performances. Figure 17 shows a sample code of the train-test split that was used. The dataset was divided into two parts: X (input features) and Y (output column). 80% of the data was used for training, while the remaining 20% was used for testing. The random state was set to 42 to ensure that the same set of samples was chosen for the training and testing dataset each time the code was executed.

5.2. Show Metrics (y_test, preds). Since this is a classification problem, the metrics under consideration are given as follows:

(1) Confusion matrix
(2) Accuracy
(3) Precision
(4) Recall
(5) F1 score

A brief description of each of these metrics is given as follows.

5.2.1. Confusion Matrix. A confusion matrix summarizes the prediction outcomes of a classification problem using count values divided by class. It displays both correct and incorrect predictions made by the classification model, providing information on the types of errors made. This breakdown is a solution to the limitation of relying only on



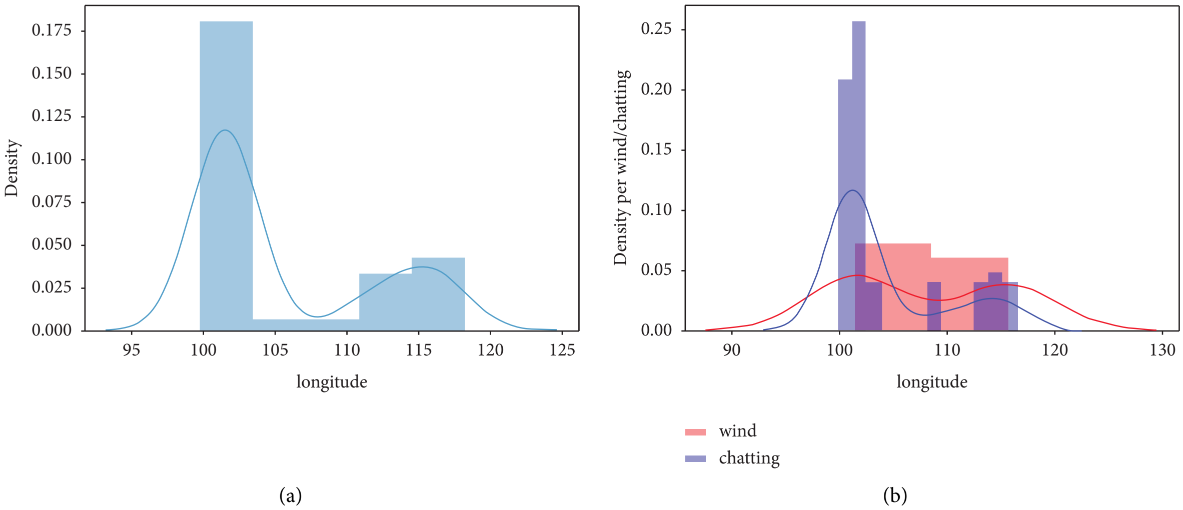
(a)

(b)

(c)

(d)

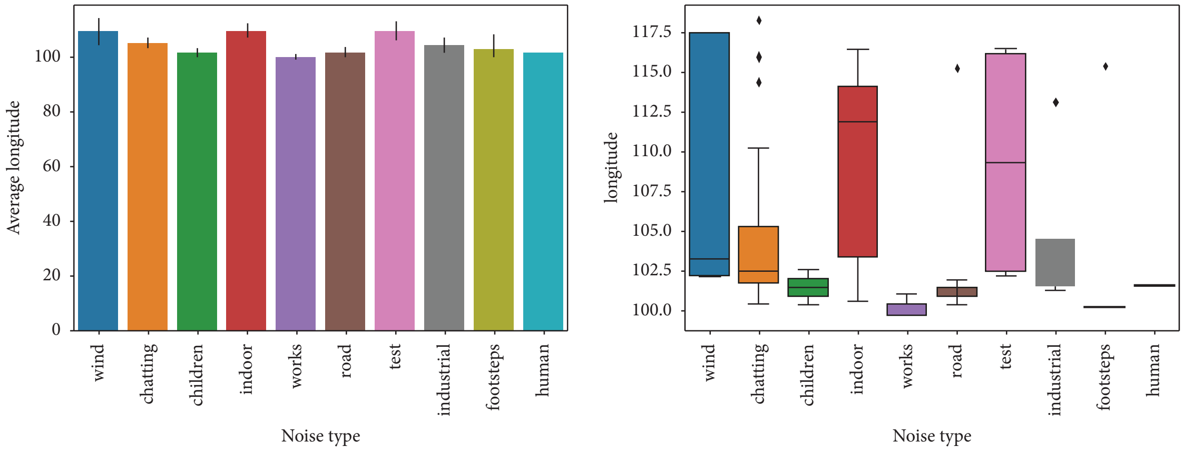
Figure 13: Plots for latitude. (a) Latitude. (b) Latitude. (c) Noise type. (d) Noise type.



(a)

(b)

Figure 14: Continued.



(c) (d)

Figure 14: Plots for longitude. (a) Longitude. (b) Longitude. (c) Noise type. (d) Noise type.

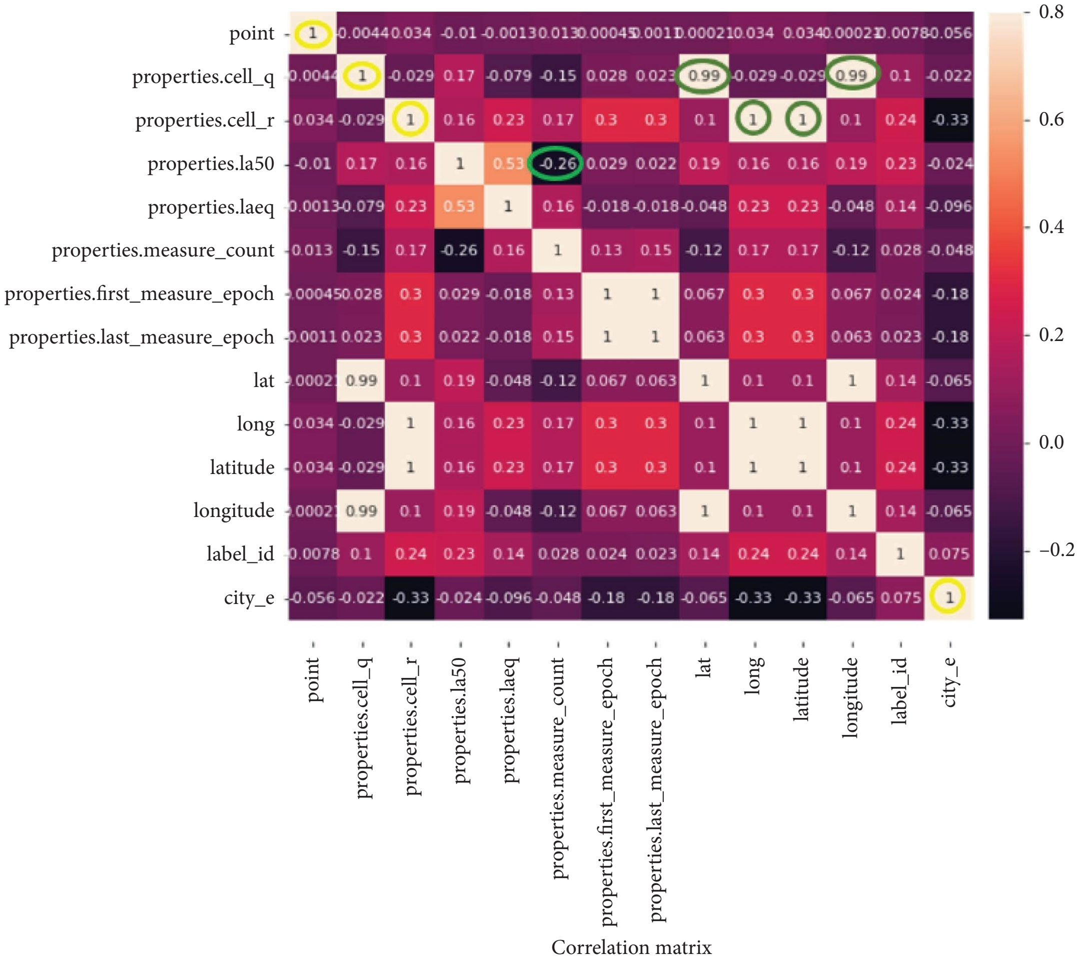
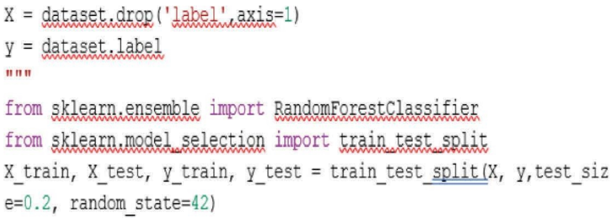


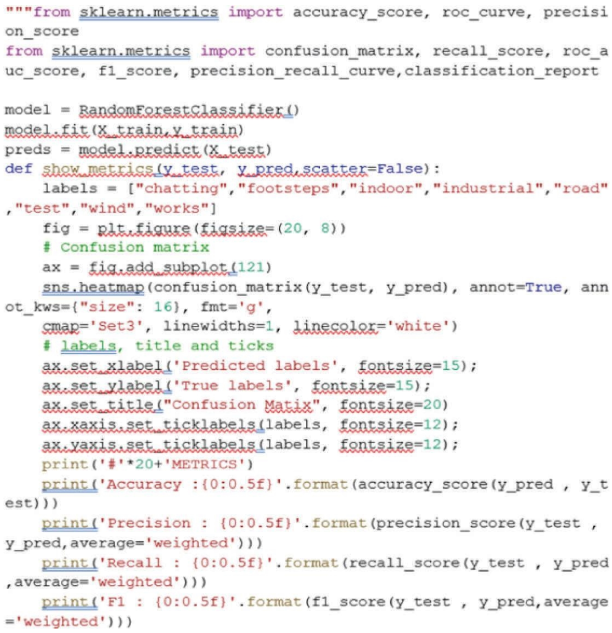
Figure 15: Correlation matrix of features.

classification accuracy. The confusion matrix is a table used to evaluate the performance of a classification algorithm. In a multiclass confusion matrix, each row represents predicted instances in a class, while each column represents actual instances in a class. The matrix's cells show how many times predicted instances were classified as actual instances. This matrix can be used to compute various evaluation metrics such as precision, recall, and accuracy.



```
X = dataset.drop('label',axis=1)
y = dataset.label
"""
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,test_siz
e=0.2, random_state=42)
```

Figure 16: Train test split code.



```
"""from sklearn.metrics import accuracy_score, roc_curve, precisi
on_score
from sklearn.metrics import confusion_matrix, recall_score, roc_a
uc_score, f1_score, precision_recall_curve,classification_report

model = RandomForestClassifier()
model.fit(X_train,y_train)
preds = model.predict(X_test)
def show_metrics(y_test, y_pred,scatter=False):
    labels = ["chatting","footsteps","indoor","industrial","road"
,"test","wind","works"]
    fig = plt.figure(figsize=(20, 8))
    # Confusion matrix
    ax = fig.add_subplot(121)
    sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, ann
ot_kws={"size": 16}, fmt='g',
    cmap='Set3', linewidths=1, linecolor='white')
    # labels, title and ticks
    ax.set_xlabel('Predicted labels', fontsize=15);
    ax.set_ylabel('True labels', fontsize=15);
    ax.set_title("Confusion Matix", fontsize=20)
    ax.xaxis.set_ticklabels(labels, fontsize=12);
    ax.yaxis.set_ticklabels(labels, fontsize=12);
    print('#'*20+'METRICS')
    print('Accuracy :{0:0.5f}'.format(accuracy_score(y_pred , y_t
est)))
    print('Precision : {0:0.5f}'.format(precision_score(y_test ,
y_pred,average='weighted')))
    print('Recall : {0:0.5f}'.format(recall_score(y_test , y_pred
,average='weighted')))
    print('F1 : {0:0.5f}'.format(f1_score(y_test , y_pred,average
='weighted')))
```

Figure 17: Performances code.

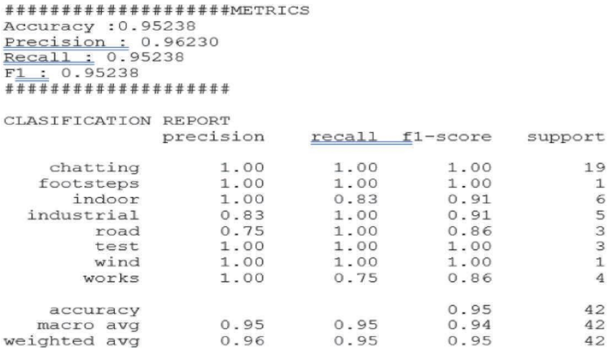
5.2.2. Classification Accuracy. As the name suggests, classification accuracy is simply the measure of how many predictions made by the model are correct (accurate). It is a relatively simple and easy to understand measure but can often be quite misleading especially when the output classes occur with varying frequencies and the output classes are not balanced in the dataset.

5.2.3. Precision. Precision is a measure of a machine learning model's performance since it measures the accuracy of a positive prediction provided by the model. Precision is calculated by dividing the number of true positives by the total number of positive predictions (i.e., the number of true positives plus the number of false positives).

5.2.4. Recall. The recall is computed as the ratio of positive samples that were properly categorized as positive to the total number of positive samples. The recall of the model assesses its ability to recognize positive samples. The more positive samples are identified, the larger the recall.

5.2.5. F1 Score. The F1 score is a crucial assessment statistic in machine learning that combines accuracy and recall to summarize a model's prediction effectiveness. Both high accuracy and recall are desirable, but there is a trade-off between the two. It is not practical to maximize both precision and recall simultaneously. As accuracy increases, recall decreases and vice versa. In Figure 18, different models are shown with sets of accuracy and recall values. The F1 score combines these two measures into a single statistic, which ranges from 0 to 1. The closer the F1 score is to 1, the better the model's performance.

5.3. Confusion Matrix. A confusion matrix is a table that is used to define the performance of a classification algorithm. In a multiclass confusion matrix, each row represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa). The cells in the matrix show that the number of times instances of a predicted class were classified as instances of an actual class. This matrix can be used to compute various evaluation metrics such as precision, recall, and accuracy. The confusion matrix in Figure 19 above illustrates the overall performance of the model. A classification report is used to measure the quality of predictions from a classification algorithm. How many predictions are true and how many are false? More specifically, true positives, false positives, true negatives, and false negatives are used to predict the metrics of



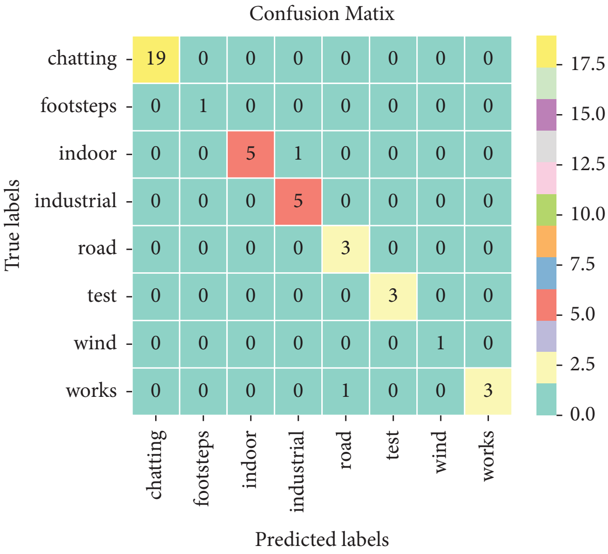
```
####################METRICS
Accuracy :0.95238
Precision : 0.96230
Recall : 0.95238
F1 : 0.95238
####################

CLASIFICATION REPORT
                 precision    recall  f1-score   support

    chatting          1.00      1.00      1.00        19
   footsteps          1.00      1.00      1.00         1
      indoor          1.00      0.83      0.91         6
  industrial          0.83      1.00      0.91         5
        road          0.75      1.00      0.86         3
        test          1.00      1.00      1.00         3
        wind          1.00      1.00      1.00         1
       works          1.00      0.75      0.86         4

    accuracy                              0.95        42
   macro avg          0.95      0.95      0.94        42
weighted avg          0.96      0.95      0.95        42
```

Figure 18: Metrics (y_test, preds) code.



	chatting	footsteps	indoor	industrial	road	test	wind	works
chatting	19	0	0	0	0	0	0	0
footsteps	0	1	0	0	0	0	0	0
indoor	0	0	5	1	0	0	0	0
industrial	0	0	0	5	0	0	0	0
road	0	0	0	0	3	0	0	0
test	0	0	0	0	0	3	0	0
wind	0	0	0	0	0	0	1	0
works	0	0	0	0	1	0	0	3

Figure 19: Confusion matrix.

a classification report. Furthermore, the confusion matrix in Figure 19 illustrates the overall performance of the model. The classification of road and industrial noises is less accurate than other categories. The goal is to maximize the values on the diagonal axis, as shown in red. The diagonal elements have the same label on the x- and y-axes, indicating that the predicted label matches the true label. The value circled in yellow is correctly predicted, as it has the label "works" on the y-axis and "industrial" on the x-axis. The scale on the left side of the matrix displays the maximum and minimum values, which range from 0 to 20. Values above 17.5 are displayed as green, and the first value of the matrix exceeding 17.5 is marked in the yellow box.

5.4. Relation between Cities and Noise Types. To investigate the relationship between each noise type and the city where the noise is captured, a bar plot of the two variables is plotted as shown in Figures 20 and 21. These figures illustrate the relationship between each noise type and each city. It is possible that the type of noise captured may be related to the city in which it is recorded.

Figure 20 shows the noise type distribution for each city. As it can be seen, the chatting noise type occurs in almost all of the cities, and hence, it can be inferred that this type of noise occurs with the highest frequency.

Figure 21 gives an overview of each noise type with properties in cell R. The works noise occurs with the highest frequency as can be seen from the histogram "correlation matrix" in Figure 22 as follows.

6. Algorithms

This research falls under the category of classification problem that can be addressed by using classification algorithms for supervised learning. Therefore, five distinct supervised learning algorithms were used to classify the data. The results of these algorithms are discussed in the sections mentioned previously.

7. Prediction Models

The evaluation of classification performance relies heavily on evaluation metrics, with the most common measure being accuracy. The accuracy of a classifier on a particular data set

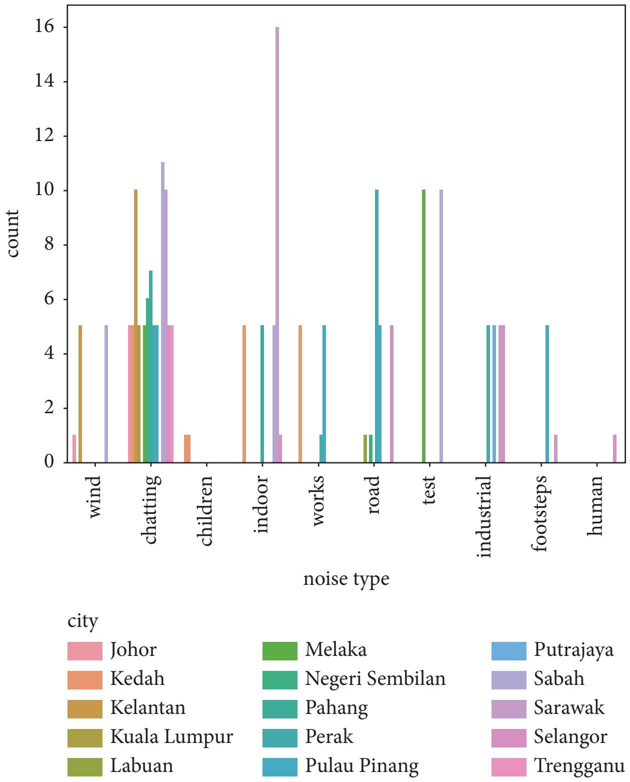


Figure 20: Noise type vs. city.

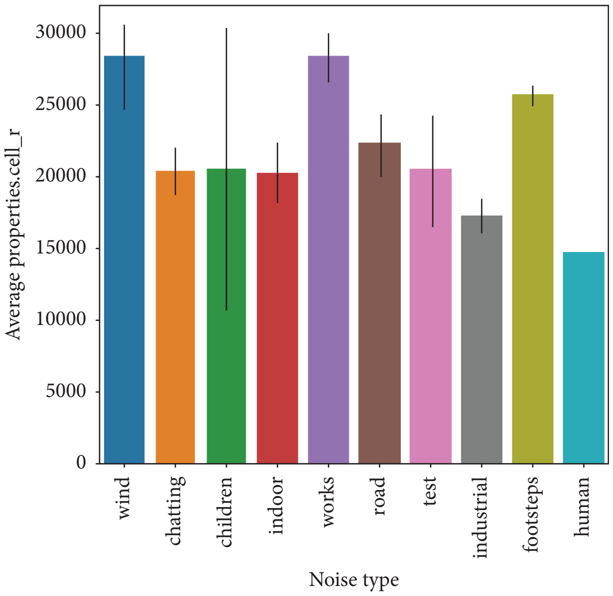


Figure 21: Noise type vs. cell R.

is represented by the percentage of test data sets that were correctly identified. However, since accuracy is often insufficient in a text mining approach, additional metrics are necessary to assess the classifier's performance. In this context, a confusion matrix is a crucial measurement tool. A confusion matrix is a combination of the following measures:

(i) TP (True Positive) represents a number of data correctly classified

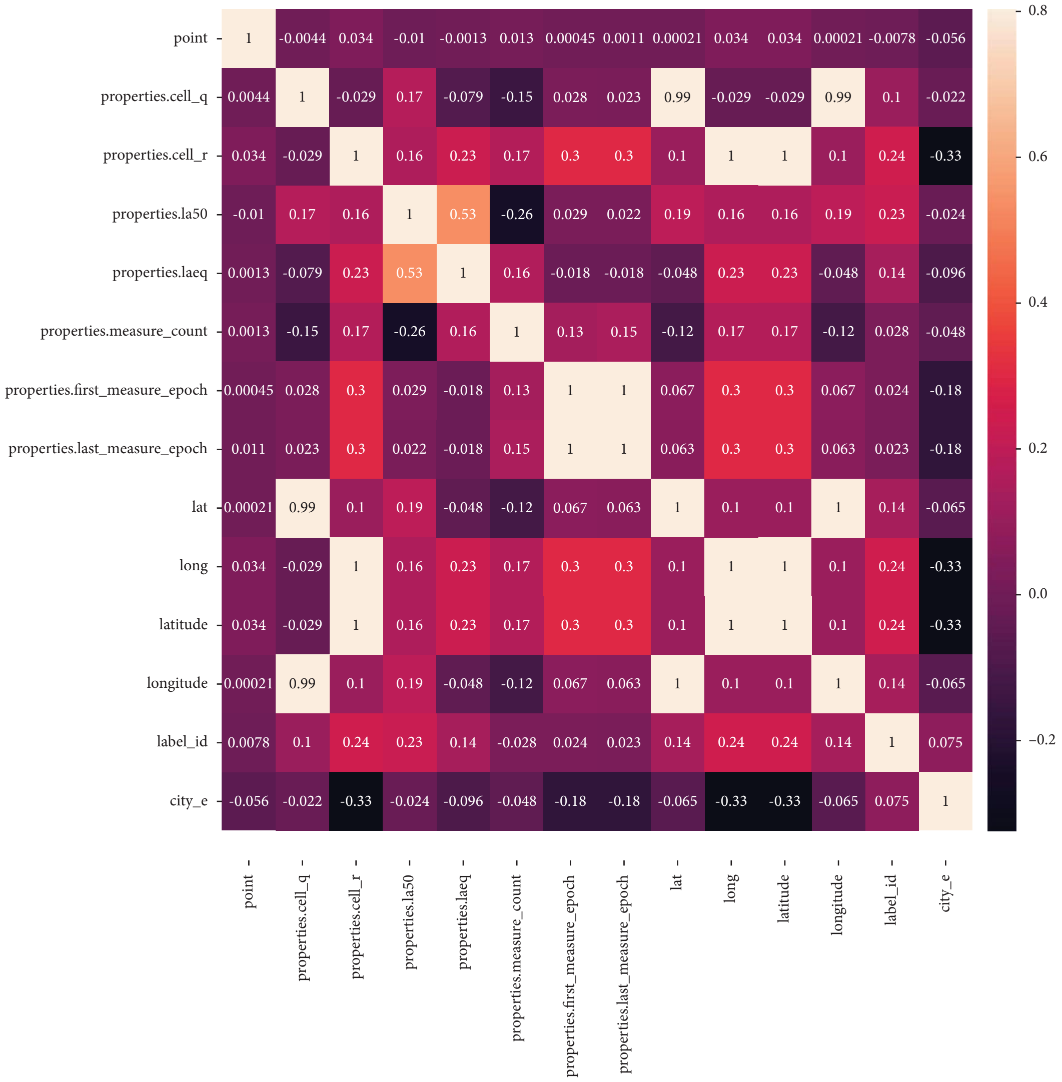


FIGURE 22: Correlation matrix.

(ii) FP (False Positive) represents the number of correct data misclassified

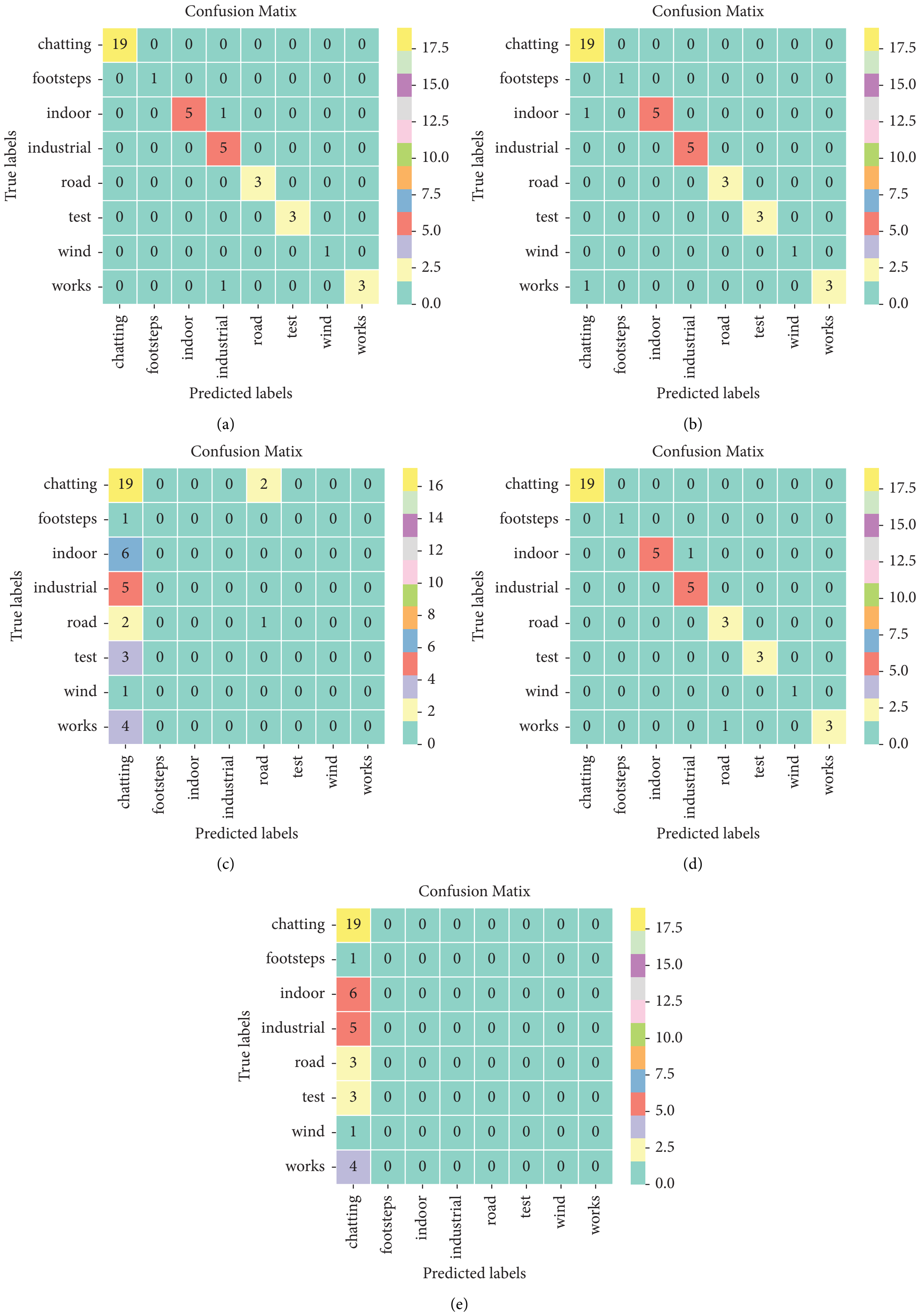
(iii) FN (False Negative) represents numbers of incorrect data classified as correct

(iv) TN (True Negative) is the number of incorrect data classified the results related to those metrics are presented in the next section

A model is created to predict the label of a given record. After the model is trained, test records are used to evaluate the model's performance. For these test records, the actual labels (also known as true labels) are known, and the model predicts a label (i.e., the predicted label). Ideally, the predicted label should be the same as the true label, but this is not always the case. In the results, such as in the decision tree, this scenario is present. The number on the diagonal represents the correctly classified noises from the test set. The F1 score is calculated from the confusion matrix, and a comparison is performed for all models. The random forest was found to be the best model.

In Figure 23, the resulting confusion matrices of each algorithm are illustrated. It can be seen that the performance of decision trees, KNN, and the random forest is better than the rest as they have predicted most types of noises correctly. The F1 score is a measure of a model's accuracy that balances precision and recall. In a multiclass classification problem, where there are more than two classes, each class is treated as a binary classification problem. The F1 score for each class is calculated, and then, an average is taken for all the classes to get the multiclass F1 score. It ranges between 0 and 1, where 1 is the best score possible and 0 is the worst. It is particularly useful when comparing models with an imbalanced class distribution. Moreover, Figure 23 also illustrates the resulting confusion matrix predictions of each algorithm. It can be seen that the performance of decision trees, KNN, and the random forest is better than the rest as they have predicted most types of noises correctly.

The F1 score as shown in Figure 24 for each class is calculated using the following formula:



Confusion Matix

	chatting	footsteps	indoor	industrial	road	test	wind	works
chatting	19	0	0	0	0	0	0	0
footsteps	0	1	0	0	0	0	0	0
indoor	0	0	5	1	0	0	0	0
industrial	0	0	0	5	0	0	0	0
road	0	0	0	0	3	0	0	0
test	0	0	0	0	0	3	0	0
wind	0	0	0	0	0	0	1	0
works	0	0	0	1	0	0	0	3

(a)

Confusion Matix

	chatting	footsteps	indoor	industrial	road	test	wind	works
chatting	19	0	0	0	0	0	0	0
footsteps	0	1	0	0	0	0	0	0
indoor	1	0	5	0	0	0	0	0
industrial	0	0	0	5	0	0	0	0
road	0	0	0	0	3	0	0	0
test	0	0	0	0	0	3	0	0
wind	0	0	0	0	0	0	1	0
works	1	0	0	0	0	0	0	3

(b)

Confusion Matix

	chatting	footsteps	indoor	industrial	road	test	wind	works
chatting	19	0	0	0	2	0	0	0
footsteps	1	0	0	0	0	0	0	0
indoor	6	0	0	0	0	0	0	0
industrial	5	0	0	0	0	0	0	0
road	2	0	0	0	1	0	0	0
test	3	0	0	0	0	0	0	0
wind	1	0	0	0	0	0	0	0
works	4	0	0	0	0	0	0	0

(c)

Confusion Matix

	chatting	footsteps	indoor	industrial	road	test	wind	works
chatting	19	0	0	0	0	0	0	0
footsteps	0	1	0	0	0	0	0	0
indoor	0	0	5	1	0	0	0	0
industrial	0	0	0	5	0	0	0	0
road	0	0	0	0	3	0	0	0
test	0	0	0	0	0	3	0	0
wind	0	0	0	0	0	0	1	0
works	0	0	0	0	1	0	0	3

(d)

Confusion Matix

	chatting	footsteps	indoor	industrial	road	test	wind	works
chatting	19	0	0	0	0	0	0	0
footsteps	1	0	0	0	0	0	0	0
indoor	6	0	0	0	0	0	0	0
industrial	5	0	0	0	0	0	0	0
road	3	0	0	0	0	0	0	0
test	3	0	0	0	0	0	0	0
wind	1	0	0	0	0	0	0	0
works	4	0	0	0	0	0	0	0

(e)

Figure 23: Prediction tables of DT, KNN, LR, RF, SVM: (a) decision tree, (b) KNN, (c) logistic regression, (d) random forest, and (e) SVM.

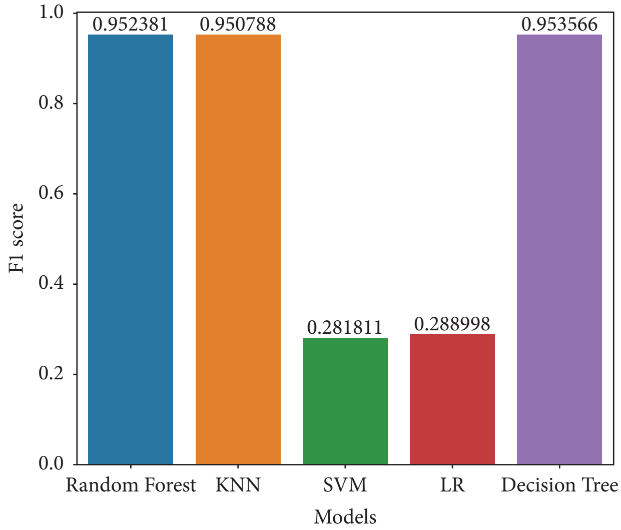


Figure 24: F1 score.

$$F1 = 2 * \frac{(\text{precision} * \text{recall})}{(\text{precision} + \text{recall})}, \qquad (1)$$

In calculating the multiclass F1 score, precision is obtained by dividing the number of true positives by the sum of true positives and false positives, while recall is obtained by dividing the number of true positives by the sum of true positives and false negatives. To determine the multiclass F1 score, the F1 score for each class is calculated, and the average of all the class F1 scores is then taken. The average can be determined by taking the mean or the harmonic mean of all the F1 scores. The harmonic mean is a better option when some classes have a low F1 score because it gives more weight to the lower values.

8. Models Comparison

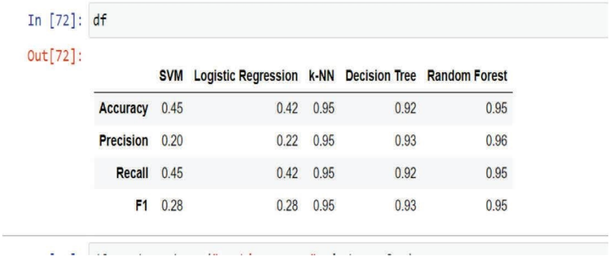
After running all models, we got the following matrices for a different model in terms of their accuracy, precision, recall, and F1 scores as shown in Figures 25 and in 26.

By creating a null hypothesis (H_0) and an alternative hypothesis (H_1) based on our problem (comparing classifiers).

H_0: the classifiers are equal

H_1: the classifiers are different

Our comparison is made using two tests: Friedman and Nemenyi. Friedman is the first test and if H_0 is rejected (H_1 is accepted), Nemenyi is used to know the best classifier. Here, we have five classifiers, SVM, logistic regression, K-NN, decision tree, and random forest. To make the Friedman test, we choose four evaluation metrics to be our reference. Should we reject H0 (i.e., is there a difference in the means) at the 95.0% confidence level? True. We will reject the null hypothesis (H_0). So, we will proceed for Nemenyi to know the best classifier (Rank).



```
In [72]: df
Out[72]:
           SVM  Logistic Regression  k-NN  Decision Tree  Random Forest
 Accuracy  0.45                0.42  0.95           0.92           0.95
Precision  0.20                0.22  0.95           0.93           0.96
   Recall  0.45                0.42  0.95           0.92           0.95
       F1  0.28                0.28  0.95           0.93           0.95
```

Figure 25: Models comparison.

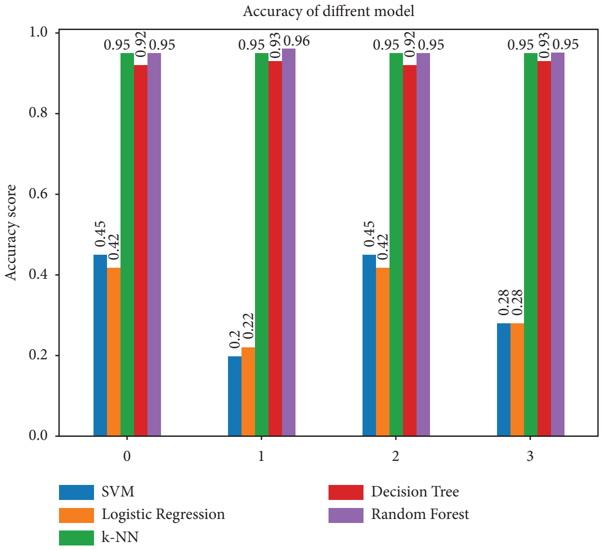
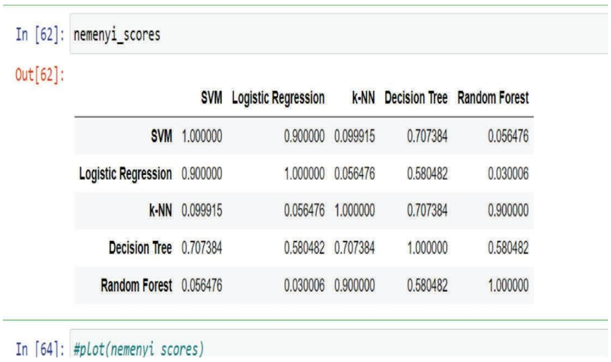


Figure 26: Accuracy comparison.

In the Nemenyi test, we need to get the difference between mean rankings (average row of ranking table) among all the classifiers (comparing pairs of classifiers). We got the following table for Nemenyi scores as shown in Figure 27.

To our classifiers, random forest and K-NN obtained higher rankings. The table shows the Holm-adjusted p values and significance (sig) for the comparisons between random forest and other models. The comparison between random forest and logistic regression yielded a Holm-adjusted p value of 0.014602, indicating a statistically significant difference. Similarly, the comparison between random forest and SVM resulted in a Holm-adjusted p value of 0.021871, also indicating a statistically significant difference. On the other hand, the comparison between random forest and decision tree produced a Holm-adjusted p value of 0.292201, which is not statistically significant. This suggests that there is no substantial difference between random forest and decision tree. Likewise, the comparison between random forest and k-NN resulted in a Holm-adjusted p value of 0.823063, indicating no statistically significant difference. Therefore, based on this comparison, we cannot confidently claim that random forest is superior to k-NN. Overall, based on these Holm-adjusted p values, random forest demonstrates statistical superiority over logistic regression and

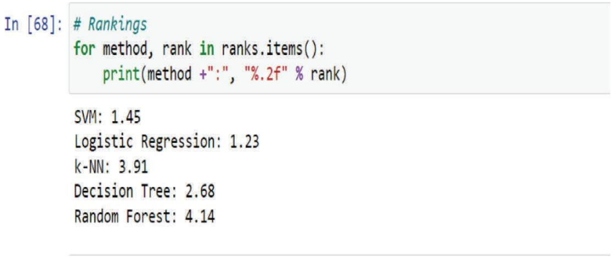


```
In [62]: nemenyi_scores
Out[62]:
```

	SVM	Logistic Regression	k-NN	Decision Tree	Random Forest
SVM	1.000000	0.900000	0.099915	0.707384	0.056476
Logistic Regression	0.900000	1.000000	0.056476	0.580482	0.030006
k-NN	0.099915	0.056476	1.000000	0.707384	0.900000
Decision Tree	0.707384	0.580482	0.707384	1.000000	0.580482
Random Forest	0.056476	0.030006	0.900000	0.580482	1.000000

```
In [64]: #plot(nemenyi_scores)
```

Figure 27: Nemenyi scores.



```
In [68]: # Rankings
         for method, rank in ranks.items():
             print(method +":", "%.2f" % rank)

         SVM: 1.45
         Logistic Regression: 1.23
         k-NN: 3.91
         Decision Tree: 2.68
         Random Forest: 4.14
```

Figure 28: Ranking scores.

SVM, but there is no significant difference between random forest and decision tree or k-NN. Based on our analysis, our classifiers, random forest and K-NN, achieved higher rankings. We discovered significant differences when comparing our top-ranked model, random forest, with the remaining models, logistic regression, and SVM. Therefore, we can confidently conclude that random forest outperforms both logistic regression and SVM. We found that logistic regression and SVM are significantly different, and we can say the random forest is better than logistic regression and SVM as seen in Figure 28.

9. Discussion and Analysis

A comparative analysis was conducted, revealing that K-nearest neighbors (KNN), random forest, and decision tree algorithms performed the best in terms of accuracy, as indicated by the confusion matrix. The F1 score, which is a measure of a model's accuracy that balances precision, recall, and speed, was evaluated. According to the figure, DT has the best speed, surpassing RF and KNN by 0.12% and 0.29%, respectively. The F1 scores of different models were assessed, and it was discovered that RF, KNN, and DT obtained the highest scores, around 0.95. On the other hand, LR and SVM had relatively lower F1 scores of 0.28 and 0.25, respectively.

10. Conclusion

It is evident from the study that noise pollution is a pressing issue that should be taken into account when planning townships or developing smart cities. Hence, this study investigates the environmental sound for noise pollution assessment. The study presents six different types of parameters used for monitoring 16 types of noise pollution data. Machine learning (ML) algorithms, including RF, KNN, DT, SVM, and LR, are used for noise monitoring and classification in Python. The results show that Python machine learning prediction results are more accurate. A comparison is made for all models from the correlation matrices results calculated. The random forest (RF) model is found to be the best with an accuracy of 100% and the fastest computation time of 0.952381. RF is an ensemble of DT that can handle large amounts of data and reduces overfitting, making it a suitable choice for ML classifications. Therefore, it is recommended to use RF as the final model for this classification problem.

Data Availability

Data have been collected in 16 different cities in Malaysia with, also, all data have been stored in Tableau. Python has been used in our research.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank all those who have contributed toward making this research paper successful. The authors wish to express their gratitude to funder by the Ministry of Higher Education under FRGS, Registration Proposal no: FRGS/1/2021/TK0/UTM/02/97.

References

[1] P. Borah and D. Gupta, "Affinity and transformed class probability-based fuzzy least squares support vector machines," *Fuzzy Sets and Systems*, vol. 443, pp. 203–235, 2022.

[2] M. Marjanović, L. Skorin-Kapov, K. Pripužić, A. Antonić, and I. P. Žarko, "Energy-aware and quality-driven sensor management for green mobile crowd sensing," *Journal of Network and Computer Applications*, vol. 59, pp. 95–108, 2016.

[3] C. Kulkarni, S. Kulkarni, B. Siddhi, and M. M. Raste, "Sound and air pollution monitoring system," *IJSER*, vol. 8, no. 2, 2017.

[4] P. Patil, "Smart IoT based system for vehicle noise and pollution monitoring," in *Proceedings of the 2017 International Conference on Trends in Electronics and Informatics (ICEI)*, pp. 322–326, Tirunelveli, India, May 2017.

[5] A. Sumithra, J. J. Ida, K. Karthika, and S. Gavaskar, "A smart environmental monitoring system using internet of things," *International Journal of Scientific Engineering and Applied Science*, vol. 2, no. 3, pp. 261–265, 2016.

[6] T. Zimmerman and C. Robson, "Monitoring residential noise for prospective home owners and renters," in *Proceedings of the International Conference on Pervasive Computing*, pp. 34–49, Seattle, WA, USA, June 2011.

[7] K. Kumar and V. K. Jain, "Autoregressive integrated moving averages (ARIMA) modelling of a traffic noise time series," *Applied Acoustics*, vol. 58, no. 3, pp. 283–294, 1999.

[8] C. Prieto Gajardo, J. M. Barrigon Morillas, G. Rey Gozalo, and R. Vílchez-Gómez, "Can weekly noise levels of urban road traffic, as predominant noise source, estimate annual ones?" *Journal of the Acoustical Society of America*, vol. 140, no. 5, pp. 3702–3709, 2016.

[9] A. J. Torija and D. P. Ruiz, "A general procedure to generate models for urban environmental-noise pollution using feature selection and machine learning methods," *Science of the Total Environment*, vol. 505, pp. 680–693, 2015.

[10] F. Van Den Berg, A. R. Eisses, and P. J. Van Beek, "A model based monitoring system for aircraft noise," *Journal of the Acoustical Society of America*, vol. 123, no. 5, p. 3151, 2008.

[11] G. Zambon, H. E. Roman, M. Smiraglia, and R. Benocci, "Monitoring and prediction of traffic noise in large urban areas," *Applied Sciences*, vol. 8, no. 2, p. 251, 2018.

[12] N. Cvar, J. Trilar, A. Kos, M. Volk, and E. Stojmenova Duh, "The use of IoT technology in smart cities and smart villages: similarities, differences, and future prospects," *Sensors*, vol. 20, no. 14, p. 3897, 2020.

[13] C. D. Kulkarni, *Design and Development of a Smartphone-Based*, 2018.

[14] A. Almehmadi, "Smart city architecture for noise pollution mitigation through the internet of things," *International Journal of Computer Science and Network Security*, vol. 18, no. 7, pp. 128–133, 2018.

[15] W. Zamora, C. T. Calafate, J. C. Cano, and P. Manzoni, "Accurate ambient noise assessment using smartphones," *Sensors*, vol. 17, no. 4, pp. 1–18, 2017.

[16] T. B. de Souza, K. C. Alberto, and S. A. Barbosa, "Evaluation of noise pollution related to human perception in a university campus in Brazil," *Applied Acoustics*, vol. 157, Article ID 107023, 2020.

[17] P. W. Wessels and T. G. H. Basten, "Design aspects of acoustic sensor networks for environmental noise monitoring," *Applied Acoustics*, vol. 110, pp. 227–234, 2016.

[18] J. E. Noriega-Linares and J. M. Navarro Ruiz, "On the application of the raspberry pi as an advanced acoustic sensor network for noise monitoring," *Electron*, vol. 5, no. 4, 2016.

[19] F. T. Espinoza and G. B. Gavilanes, "Sound noise levels calibration with smart-phones," in *Proceedings of the 21st Euromicro Conference on Digital System Design (DSD 2018)*, pp. 1–6, Prague, Czech Republic, August 2018.

[20] I. Kirillov and V. Bulkin, "The mobile system of urban area noise pollution monitoring," in *Proceedings of the 2015 Second International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T)*, pp. 200–203, Kharkiv, Ukraine, October 2015.

[21] Y. Alsouda, S. Pllana, and A. Kurti, "A machine learning driven IoT solution for noise classification in smart cities," in *Proceedings of the 21st Euromicro Conference on Digital System Design (DSD 2018)*, Prague, Czech Republic, August 2018.

[22] V. Bountourakis, L. Vrysis, and G. Papanikolaou, "Machine learning algorithms for environmental sound recognition: towards soundscape semantics," in *Proceedings of the Audio Mostly 2015 on Interaction with Sound*, pp. 1–7, Thessaloniki Greece, October 2015.

[23] C. Sparke, "Environmental noise classification through machine learning," *Proceedings of ACOUSTICS*, vol. 7, p. 9, 2018.

[24] F. Demir, D. A. Abdullah, and A. Sengur, "A new deep CNN model for environmental sound classification," *IEEE Access*, vol. 8, pp. 66529–66537, 2020.

[25] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, Berlin, Germany, 2006.

[26] A. O. Albaji, R. B. A. Rashid, M. A. B. Sarijari, Z. B. Salam, S. Z. A. Hamid, and Y. H. Ali, *A Machine Learning for Environmental Noise Monitoring and Classification Using Matlab*, Universiti Teknologi Malaysia, Johor, Malaysia, 2021.

[27] Y. H. Ali, R. A. Rashid, and S. Z. A. Hamid, "A machine learning for environmental noise classification in smart cities," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 3, pp. 1777–1786, 2022.

[28] U. Mishra, D. Gupta, and B. B. Hazarika, "An intuitionistic fuzzy random vector functional link classifier," *Neural Processing Letters*, vol. 06, pp. 1–22, 2022.

[29] B. B. Hazarika and D. Gupta, "Affinity based fuzzy kernel ridge regression classifier for binary class imbalance learning," *Engineering Applications of Artificial Intelligence*, vol. 117, Article ID 105544, 2023.

[30] B. B. Hazarika and D. Gupta, "Random vector functional link with ε-insensitive Huber loss function for biomedical data classification," *Computer Methods and Programs in Biomedicine*, vol. 215, Article ID 106622, 2022.