Feasibility study for the automation of environmental noise contamination detection for community noise fly-over testing

Todd Schultz, Ph.D.,[1]
The Boeing Company, Seattle, WA, 98108, USA

Rahul Birmiwal,[2]
University of Washington, Seattle, WA, 98195, USA

and
Sean Miller[3]
Bungie, Inc., Bellevue, WA, 98004, USA

A feasibility study to replace human listeners with an automated machine learning-based approach for detecting environmental noise contamination for aircraft noise certification fly-over testing is presented. The potential business impact that could result is a large cost savings with the reduction of testing personnel, equipment, and improved testing efficacy. This study leverages decades of research work using acoustic signal features and classification methods from other industries to compose a wide variety of features to test such as octave spectrum, wavelets and classifiers such as logistic regression, neural networks, and long short-term memory networks. The results show that many feature set and classifier combinations can achieve greater than 90% accuracy. Three particular combinations are optimized and tested against randomized simulated data. Two of the three are based on discrete wavelet transforms and the other is based on a modified mel-frequency cepstral coefficient feature set. All three achieve similar performance of 93% accuracy but decreased to approximately 81% to 83% when tested with randomized simulated data with variation in the signal-to-noise ratio and the proportion of the signal block contaminated. Additional audio recordings and simulated signal blocks for improving the robustness of the training data is recommended for improving the generalizability of the feature set and classifier combinations.

I. Introduction

Community noise flight testing for aircraft noise certification requires extensive equipment and personnel to achieve efficient and effective results. The certification requirements are regulated by government agencies such as the Federal Aviation Administration (FAA) in the United States with the Code of Federal Regulations Title 14, Part 36 [1], the European Aviation Safety Agency in the European Union with International Civil Aviation Organization (ICAO) Annex 16 [2], and the Civil Aviation Administration of China in the People's Republic of China. The required flight testing involves flying a test aircraft at low altitude over a test site, typically a rural airport. A site at the end of the runway is instrumented with acoustic sensors to record the noise levels of the aircraft during a flyover. The noise data are recorded for various configurations comprised of different airframe configurations and engine power settings to simulated take-off, approach, and landing noise levels. The noise data are processed for comparison to the allowable limits established by the regulations where exceeding the limits risks failing the certification test.

To ensure the best possible data for the noise certification tests, aircraft companies seek test sites with low background or ambient noise levels such as The Boeing Company test facility at Glasgow Industrial Airport (FAA LID: 07MT) as seen in Fig. 1 [3]. In addition, test personnel may be assigned to monitor the microphone signals for noise contamination that would increase the measured and recorded noise levels of the aircraft. Typical sources of the environmental noise contamination include bird chirps and other wildlife/livestock vocalizations, insect noises, the sound of traffic, and aircraft sounds from flights other than the test airplane. The noise monitors can alert testing staff

[1] Test & Evaluation Engineer, Boeing Test & Evaluation, and AIAA Member.
[2] Graduated student, Master of Science in Data Science.
[3] Data Engineer, SAI.

to the presence of noise contamination such that corrective action can be taken. These actions include voiding the condition and requesting a repeat run, requesting a delay in the arrival of the test aircraft, and proactive removal of biological sources. This approach is costly due to:

- Travel costs for the monitors.
- Specialized workstation computers and software for the monitoring workstations.
- Storage, shipping, setup, and networking support for the workstation computers.
- Validation and testing of the analysis applications, especially the communications with other data acquisition systems at the test site.

Fig. 1 Aircraft noise fly-over at The Boeing Company test facility near Glasgow, MT (circa 2012).

Also, human monitors can be subjected to:

- Mental fatigue of the monitors from the repetitive menial task, resulting in a reduction of the quality and consistency of the classification over time.
- Inconsistencies and subjectivities of the classification from monitor to monitor.
- Limited ability or knowledge to accurately account for noise contamination corrections available to analysis staff at post-acquisition data reduction.
- Limited post-test review opportunities for training and enhancements.

An automated system is desired that can minimize the costs and overcome the limitations of the existing solution. The goal of the automated monitoring system would be to detect, or classify, the presence of environmental noise contamination in the acoustic signals as they are acquired. Furthermore, from that classification, it would also provide guidance whether the contamination has corrupted the data, whether the flight condition should be repeated, or to react and remove the noise sources before the airplane is on-condition and thus avoid repeating the condition. The work presented here is limited to the first step in the creation of such an automated system by studying the feasibility of machine learning algorithms to detect biological noise contamination in recorded acoustic signals that contain only ambient noise or aircraft noise. Thus, this work is limited to creating and evaluating different feature sets derived from the signals and supervised classification algorithms on recorded data. The fully developed automated system would remove the need for the multiple work stations and staff providing significant cost reductions for the community noise fly-over capability. The automated system also could provide increased accuracy and consistency of the classification thus increased efficacy of the test for further significant cost reductions. The detector should alert community noise test crews of the presence of environmental noise contamination continually, in real-time, allowing them to respond by either removing the sources from the measurement area before the arrival of the airplane or by declaring the on-condition recording out of tolerance and requesting a repeat of the condition.

The remainder of this paper is outlined as follows. The next section describes the example aircraft flyover noise and contamination noise recordings used in this feasibility study. Then the following section describes overall process

of creating a machine learning algorithm for contamination detection including the signal processing algorithms for feature generation and lists the machine learning algorithms investigate here. Then a discussion of the results of detection algorithms on the example data is provided. Finally, this paper finishes with the conclusions derived from the results and recommendations for additional work.

II. Data

Boeing Test & Evaluation provided 49 aircraft flyover signatures, 7 contamination signatures, and 10 ambient signature recordings in standard wave file format. The recordings were collected near a major commercial airport in the USA over three separate days during the winter of 2017-2018. All data were recorded on scientific grade instrumentation with a usable bandwidth from 5 Hz to 20 kHz with a sampling rate of 51.2 kHz as shown in Fig. 2. Aircraft consisted of turboprop and turbofan aircraft used for commercial flights such as the Bombardier Q400 turboprop, the Boeing 737, the Airbus A350, and others. All aircraft signature recordings were individually reviewed and trimmed to only include the portion of the recording with an audible aircraft noise signature without any contamination. The ambient recordings were individually reviewed to ensure no contamination or other noise sources were present. The contamination signals were individually reviewed and trimmed to only include the portion of the recordings with an audible contamination signature such as a bird call or road noise and was absent of any aircraft noise. All recordings from Boeing Test & Evaluation were anonymized by an unknown normalization factor to remove the absolute noise levels and physical engineering units from the recordings and the recordings were labeled with only the following categories: turboprop, turbofan, contamination, and ambient. Fig. 3 shows an example of an aircraft fly-over audio recording and an example of a bird vocalization audio recording is shown in Fig. 4. The goal of this feasibility study is to automatically detect if there is a signal such as the bird vocalization present in the recorded aircraft signature.

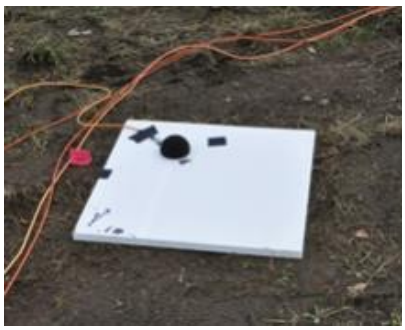

Fig. 2 Example of the microphone installation for the Boeing Test & Evaluation audio recordings.

Six additional contamination recordings of wildlife vocalizations from the National Parks Service (NPS) provided as public domain data are added to the data set to provide a greater variety of contamination examples [4]. The original recordings from the National Parks Service are provided in the mp3 file format. The recordings included crickets, bison, common raven, sparrow, Steller's jay, and yellow-rumped warbler vocalizations and were simple audio recording with no known information about the calibration of the recording instrumentation. The Steller's jay signature was recorded at a sampling rate of 16 kHz and the remaining five were recorded at 44.1 kHz. All the NPS data is resampled to match the sampling rate of the aircraft signatures provided from Boeing Test & Evaluation. In total, there are 59 uncontaminated audio recordings and 13 audio recordings of possible contamination sources available for this study.

III. Methods

There are many methods available to generate feature sets from audio data and many machine learning algorithms to classify a set of features [5], [6], [7], [8]. To follow the broad purpose of this feasibility study, a wide variety of both are consider and described. The analysis followed the general process of loading each clean audio recording, splitting the audio signals into records or blocks and processing each block into features, loading the contamination audio records, splitting them into blocks, superimposing the contamination to the clean audio signals, processing the contaminated audio into features, and finally training the classification algorithm on the combined clean and contaminated feature set. The performance of the feature set and classification algorithm is evaluated on its classification accuracy using a cross-validation method. The classification accuracy is the percentage of predicted classes the algorithm correctly identified as compared to the known, true classes. Using cross-validation ensures that

the algorithms performance isn't measured with observations seen by the algorithm during training, thus improving the generalizability of the performance score [9].

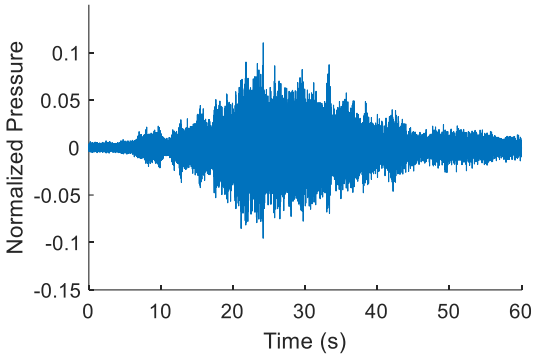

Fig. 3 An example of an audio recording for an aircraft fly-over.

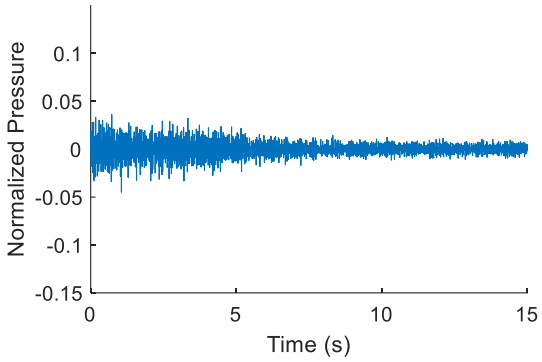

Fig. 4 An example of an audio recording for a bird vocalization.

A. Signal Processing

The audio recordings are used to generate a data set suitable for types of signal processing and feature extraction investigated in this study and appropriate for classification machine learning algorithms. The 59 uncontaminated audio recordings are used as the set of available data representative of either uncontaminated aircraft signatures or uncontaminated ambient noise. The 13 audio recordings of potential contamination sources are used to represent acoustic contamination on the desired signals and are used to create a set of 59 contaminated signatures through the use of superposition with the uncontaminated recordings. The audio recordings are prepared for feature extraction by the following procedure. First, parameter values for the block length, block overlap, and the signal-to-noise ratio between the uncontaminated signal and contamination signal are prescribed for the test data set. Then an uncontaminated audio recording is loaded from the file. Next, the average power in the uncontaminated signal is normalized such that it is unity. Next, the signature is separated into overlapping blocks of an assigned length and overlap. Each individual block of the signal is now treated as an observation of the signal itself and will create an individual set of features for the prediction algorithms. Next, the audio recording has one of the contamination recordings randomly assigned to it, without substitution. The contamination signal is loaded from the file and the length compared to the uncontaminated signal. As needed, the contamination signal is repeated and trimmed to match the length of the uncontaminated signal. Next, the contamination signal is normalized to unity average power and segmented into blocks identically to the uncontaminated signal. Each block of the contamination signal is added to the corresponding block of the uncontaminated signal with an adjustment of the relative average power in the contamination signal block to achieve the prescribed signal-to-noise ratio between the uncontaminated signal and the contamination signal. This is then repeated for each of the 59 uncontaminated audio recordings and generates a total of 118 sets of signal blocks with class parity between uncontaminated and contaminated signal blocks.

B. Feature Creation

One of the goals of this study is to investigate the prediction performance of various methods for creating the feature sets from the signal blocks. The following sections briefly describe the methodologies used for feature generation.

1. Octave Spectral Features

Octave or constant percentage bandwidth spectral processing is used as a basic method for generating features. The advantages of this method are its simplicity and that the features are common signal analysis results descripting the spectral energy contained in the signal. Since, the average power is normalized for the signals themselves, the estimated power in the octave bands are only useful for relative measured compared to each other and such the total spectral power of the estimated octave spectrum of each signal block is normalized to unity. Two octave-based features set are generated in this study. The first is the full octave feature set starting with the octave band centered at 31.6 Hz and ending with octave band centered at 15.849 kHz for a total of ten features. The second is the third octave feature set starting with the band centered at 31.6 Hz and ending with the band centered at 19.953 kHz for a total of 29 features. Each signal block of the data is treated as an individual input and results in an individual set of feature or observation.

2. *Mel-Frequency Cepstral Coefficients*

Mel-frequency cepstral coefficients (MFCC) is a method to analyze a signal with cepstral coefficients or log energy coefficients that have frequency bands that are equally spaced on the mel-frequency scale [7], [10]. The mel-frequency scale is designed to approximate the human auditory response and can lead to improved signal representations as compared to a linear frequency scale. The frequency range of the MFCC is from 133 Hz to 6.845 kHz and the number of features generated can vary [10], [11]. This study only considered MFCC feature sets with 13 features and 26 features.

3. *Modified Mel-Frequency Cepstral Coefficients*

The modified mel-frequency cepstral coefficients extract the cepstral coefficients or log energy coefficients similar to the MFCC but with the upper frequency range extended to 20 kHz. This is done as the original mel-frequency cepstral coefficient processing is designed around human speech and the modified mel-frequency cepstral coefficients are designed to record higher frequency content that may be present in general audio recordings. The audio signals of interest in this problem are from turbo machinery and wildlife vocalizations which can exceed the human speech frequency range.

4. *Continuous Wavelet Transform*

The continuous wavelet transform (CWT) is a process that is used to decompose a signal into wavelets. A scalogram is a visual representation of a CWT with the axes of time and scale where each pixel in the image is colored or shaded by the magnitude of the coefficient. A scalogram is similar to a traditional Fourier-based spectrogram. For this study, two feature sets are generated from CWT using the Morse wavelet: the clean and contaminated scalogram images and a bag of image features using speeded up robust features (SURF) [12], [13]. The SURF features generated from this process are clustered using k-means to reduce the encoding to 500 features. One positive of scalograms as a feature set is their ease of interpretability as noise contamination tends to stand out as seen in Fig. 5.

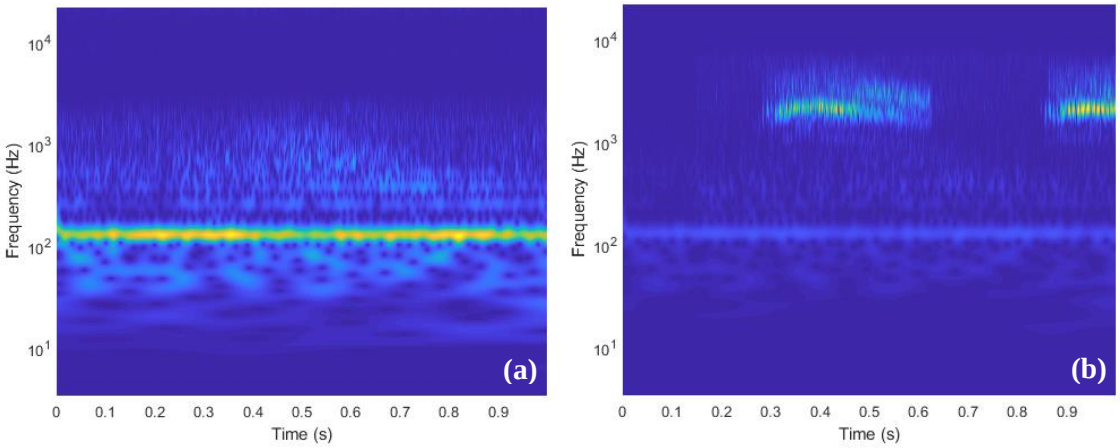

Fig. 5 Scalogram images. (a) with only aircraft signal and (b) with the aircraft signal and contamination.

5. *Discrete Wavelet Transform*

The discrete wavelet transform (DWT) decomposes signal into correlations with a shift and scaled versions of a sampled mother wavelet. Unlike the CWT where the scaling parametrization is flexible, the DWT requires scaling in powers of 2 resulting in often a sparser, more compact representation of the spectral energies in the signal. Like the CWT, the DWT has gained considerable interest in a variety of signal processing and feature extraction domains, due to its superiority in "tiling" the time-frequency joint space over other transforms such as the short-time Fourier transform where the tiling is uniform [14]. A second and equally significant reason is because a multi-level DWT is fast implementable using a "binary tree" filter-bank recurrence algorithm that yields the approximation and detail coefficients of the transform by convolutions of the original signal with two quadrature mirror filter banks appropriately created by discrete samples of the continuous mother wavelet that satisfy certain orthonormality conditions. The chosen discretizing scaling defines the "tiling" and yield a frequency multiresolution lying on a dyadic scale, making the DWT especially suited for acoustic applications involving human hearing [15], [16], [17].

6. *Discrete Wavelet Packet Transform*

The discrete wavelet packet transform (DWPT) is similar to the DWT, with the primary difference that the tree-branching convolution with the filters recurs on both the approximation and detail coefficients, whereas the DWT

only operates on the approximation coefficients. Thus, the DWPT yields a more balanced sub-band coding decomposition [16].

A feature set was produced using three feature extraction methods based upon the DWT and DWPT and concatenating the resultant values to form a feature vector for each signal block in the training set. These methods are, for an *L*-level DWPT and DWPT and adapted from work described in [18].

1. The Shannon entropy matrix formed by performing the DWPT on each signal block, calculating the Shannon entropy of the transform and concatenating entropy values at the terminal node (leaves) coefficients [19].
2. The "wavelet variance" [20] of the maximal-overlap computation method for the DWT. This effectively yields a compact octave-band representation of the signal which captures the time-varying frequency characteristics per octave into a single value per passband.
3. A set of inter and intra-level coefficient statistics as suggested by [21].

The primary consideration of the DWT/DWPT feature set is the level of the transform. In particular, there exists a performance-complexity tradeoff: higher-level transforms may yield finer resolution in time-frequency joint space, but at the cost of doubling computational time required per unit level increase in the transforms. This is especially true in computing the Shannon entropies of the DWPT due to the full, balanced binary tree produced using that algorithm. The size of the resultant feature vectors for a 3, 4, and 5 level transforms are 17, 29, and 49 respectively.

C. Classification Algorithms

Many classification algorithms are investigated ranging from traditional logistic regression to deep neural networks [22], [23]. Cross-validation with 5-folds is used to evaluate the performance of each classifier. The main performance metric used to measure and evaluate the classifiers is the accuracy, which is the proportion of all predictions that are correct as compared to the known class of the test data. Other metrics exist such as the F1 score [24] and are used later to enrich the evaluation.

1. Conventional Classifiers

Conventional classifiers investigated include logistic regression, support vector machines, and decision trees [22], [23]. Various kernels are studied include linear, quadratic, and cubic kernels and a family of Gaussian kernels with three different kernel scale of 14 (coarse), 3.6 (medium), and 0.9 (fine). Decision trees are investigated for three different sizes with up to 4 maximum splits for coarse, 20 maximum splits for medium, and 100 maximum splits for fine.

2. Ensemble Classifiers

Ensemble classifiers are a family of classifiers that use multiple conventional or other classifiers as weak learners to produce many outputs predict classes that are then combined to produce a single prediction using majority vote. Ensemble classifiers can be especially useful in reducing overall variance of the system and are robust to outliers [23]. This study considered three different types of ensemble classifiers: boosted trees, bagged trees (random forest), and subspace k-nearest neighbors. The boosted trees utilized an AdaBoosting algorithm with 30 simple decision tree classifiers each with 20 maximum splits and a 0.1 learning rate. The bagged trees ensemble utilized a bootstrap aggregating algorithm with 30 simple decision trees. The subspace k-nearest neighbors utilized 30 k-nearest neighbor classifiers with each only containing a subspace of seven of the original input predictor variables chosen independently and randomly for each k-nearest neighbor classifier.

3. Shallow Neural Networks

Shallow or conventional artificial neural networks are a non-linear computational model using artificial neurons to create complex functional relationships [23]. The shallow neural networks investigated are simple networks with an input layer for accepting all the features, hidden layers for learning the relationship between the inputs and the known outputs or classes, and an output layer for gathering the results of the hidden layers and providing a single output value. The shallow neural network investigated uses 10 hidden neurons in the hidden layer trained using the Levenberg-Marquardt algorithm [25].

4. Deep Neural Networks

Convolutional Neural Network

Convolutional neural networks (CNN) are a class of deep neural network designed to require minimal processing of the data beforehand, instead allowing the network to find the salient features [26]. However, CNN can be computational expensive and numerically difficult to train. A common method to avoid training is to use transfer learning were a pretrained network is used as starting point for learning on a new data set. This study used the GoogleNet network architecture which is more commonly referred to as an Inception network [27].

Table 1 Summary of the initial broad investigation of feature sets and classifiers. The signal-to-noise ratio for all contamination signals is 6 dB. MFCC is the mel-frequency cepstral coefficients, SVM is support vector machines, KNN is k-nearest neighbors, CNN is convolutional neural networks, DWT is discrete wavelet transform, LSTM NN is long short-term memory neural networks. All DWT feature sets include Shannon entropy, wavelet variance, and sub-band statistics, unless otherwise noted. The top 10% of performers are indicated by italicized numbers and by a colored background.

Feature set	# Features	Block length (s)	Logistic regression	Fine tree	Medium tree	Coarse tree	Boosted trees ensemble	Bagged trees ensemble	Linear SVM	Quadratic SVM	Cubic SVM	Fine Gaussian SVM	Medium Gaussian SVM	Coarse Gaussian SVM	Subspace KNN	Neural Net	CNN	LSTM NN
Modified MFCC	13	1	75.8%	83.1%	81.1%	75.6%	85.0%	86.8%	77.2%	87.3%	89.1%	85.7%	88.0%	80.4%				
Modified MFCC	13	2	78.3%	85.2%	82.5%	76.3%	86.3%	88.2%	79.0%	89.5%	***91.9%***	88.6%	89.3%	80.5%		86.7%		
Modified MFCC	26	1	78.0%	81.8%	78.9%	75.6%	85.0%	85.5%	79.0%	88.8%	89.7%	76.3%	88.1%	79.7%				
Modified MFCC	26	2	80.2%	85.2%	81.4%	74.3%	85.9%	88.3%	80.4%	***90.7%***	***92.8%***	80.9%	90.3%	79.7%		***92.5%***		61.8%
1/3 Octaves	29	1	70.7%	83.7%	80.2%	75.3%	85.1%	89.0%	70.3%	77.5%	81.2%	71.9%	75.4%	70.1%				
1/3 Octaves	29	2	72.8%	87.5%	82.8%	77.3%	87.5%	***91.9%***	71.7%	79.7%	83.8%	76.6%	76.5%	71.5%		71.5%		
Octaves	10	1	67.3%	82.0%	79.4%	72.7%	82.2%	86.3%	66.6%	76.4%	80.0%	78.0%	75.6%	66.8%				
Octaves	10	2	67.8%	84.7%	80.6%	73.7%	84.2%	89.0%	66.5%	77.0%	81.2%	81.1%	76.8%	66.9%		77.4%		
MFCC	13	2	76.3%	82.0%	78.2%	72.1%	83.5%	85.3%	76.8%	85.2%	87.4%	83.8%	87.0%	77.0%				
MFCC	13	1	74.1%	80.4%	78.9%	68.8%	81.6%	82.9%	75.3%	83.0%	84.6%	80.5%	84.8%	76.4%		77.5%		
FFT (25 Hz resolution)	799	0.041	61.3%	78.4%	75.8%	70.0%												
FFT (100 Hz resolution)	200	0.01	68.3%	76.8%	74.0%	69.6%												
CWT Scalogram	150528	1															***92.9%***	
CWT Scalogram	150528	3															85.1%	
CWT Scalogram	150528	5															83.3%	
CWT Scalogram Bag of Features	500	1	81.0%	75.5%	74.2%	73.3%	82.1%	82.0%	83.1%	86.4%	87.1%	60.5%	86.1%	79.9%				
CWT Scalogram Bag of Features	500	3	80.7%	73.2%	73.7%	75.3%	84.8%	82.7%	86.7%	87.7%	88.6%	59.1%	87.9%	81.6%				
CWT Scalogram Bag of Features	500	5	55.0%	73.0%	74.1%	70.6%	82.3%	81.7%	88.0%	89.4%	***90.9%***	60.2%	87.7%	81.6%				
DWT (Coiflet2, 4 levels)	29	1	70.0%	90.4%	88.3%	78.1%	***91.2%***	***93.1%***	80.1%	86.2%	88.6%	84.8%	82.3%	76.0%		81.5%		76.4%
DWT (Coiflet2, 4 levels)	29	2	75.4%	***91.6%***	88.1%	80.1%	***91.0%***	***93.3%***	80.5%	87.3%	***90.5%***	85.8%	82.4%	77.6%		81.5%		78.2%
DWT (Coiflet2, 5 levels)	49	2	81.5%	***91.7%***	89.1%	79.3%	***92.3%***	***94.2%***	82.1%	88.0%	***90.6%***	87.2%	83.5%	78.8%		82.3%		
DWT (Coiflet2, 5 levels, Hampel Filter)	49	2	81.0%	***91.7%***	88.7%	79.3%	***92.1%***	***94.0%***	82.0%	88.9%	***91.5%***	87.6%	83.8%	76.8%	76.5%	84.9%		
DWT (Coiflet2, 4 levels, Hampel Filter)	29	2	70.1%	***92.1%***	88.6%	80.0%	***91.1%***	***93.5%***	80.7%	87.9%	***90.8%***	86.3%	82.7%	77.8%		83.1%		
DWT (Coiflet2, 3 levels, Hampel Filter)	17	2	67.1%	89.7%	87.5%	78.1%	89.6%	***92.9%***	79.0%	84.4%	52.0%	85.9%	81.1%	76.5%	81.3%	81.4%		
DWT (Debauchies4, 4 levels)	29	1	64.4%	90.1%	86.0%	81.1%	88.9%	***91.6%***	77.2%	80.6%	51.4%	81.5%	77.6%	66.1%	71.1%			
DWT (Haar, 4 levels)	29	1	53.8%	77.0%	73.1%	68.7%	76.4%	80.5%	71.5%	76.4%	58.7%	76.1%	73.9%	70.4%	67.1%			
DWT (Coiflet2, 4 levels, no entropy)	25	2	71.6%	***90.6%***	85.7%	81.6%	88.3%	***92.1%***	79.2%	57.1%	49.9%	76.3%	67.9%	61.0%	54.1%			
DWT (Coiflet2, 2 levels)	7	2	70.1%	89.0%	84.6%	78.9%	88.2%	***92.3%***	68.8%	71.8%	53.9%	77.7%	74.0%	61.5%	73.7%			

Long Short-Term Memory Neural Networks

Long short-term memory neural networks are neural networks with a layer capable of learning the long-term dependencies between time samples of a sequence of inputs or features [8], [28]. Thus, the nature of the change in values from one feature set to another is also used as an additional feature. The LSTM network studied here used 100 hidden neurons and trained with a maximum of 40 epochs and only used 80 variables per an iteration.

IV. Results

The proposed feature sets and classifier algorithms are investigated in two phases as such:

1. Broad investigation of widely varying feature sets and model types.
2. Deep investigation of the top 3 models from the previous phase.

The initial broad investigation is designed to gather information regarding the general performance of various feature sets and classifier algorithms in general without extensive, detailed work. In particular, the initial phased avoids the time-consuming steps of hyperparameter optimization and robust validation testing. The goal for this phase is to narrow down and select the top two or three performing types of models that will be studied and characterized in detail in the second phase. The deep investigation phase is intended to find the optimal performance for each selected feature set and classifier combination including optimizing the hyperparameters for each classifier algorithm and stress test the models for various simulated situations to gauge potential real-world performance.

A. Initial Broad Investigation

The initial broad investigation of feature sets and classifier algorithms is executed with existing software libraries from the MathWorks (The MathWorks, Inc.) using default values to train the classifier algorithms to quickly cover a wide variety of combinations. The out-of-sample accuracy computed from cross-validation is used to compare the various combinations and find the best performers for continued study in the next phase. A summary of the results is given in Table 1. Overall, the highest accuracy achieved is 94.2% by the DWPT feature set with the Coiflet2 wavelet with a 5-level decomposition and the bagged tree classifier. The wavelet-based feature sets with the bagged tree classifier are strong performers occupying 9 positions out of 23 in the 10% of the results. The best non-wavelet-based performer is the modified MFCC with 26 features and the cubic SVM classifier with an accuracy of 92.8%, the seventh best overall score. However, the computational time to compute the Shannon entropy feature in the DWPT feature sets is noted to be the most time-consuming and may impact the ability of the those feature set to be executed in real-time. A 2 second block or 102,400 samples of data required 0.94 seconds to compute the full DWPT feature set with a 5-level wavelet decomposition on a mid-level desktop computer (Intel Core i5-6600, 16 GB RAM) which may impact the implementation of the algorithms into a real-time monitoring system for the intended flyover community noise application. A 4-level wavelet decomposition for the full DWPT feature set only half the computational time and may be more practical for final implementation. Note that the computational time to train each of the classifiers will not affect the practical application of method as training is not performed in-situ during the flyover tests.

The feature set and classifier combinations selected for the next phase of the study are given in Table 2. The DWT5 feature set/classifier is chosen as it is scored the highest overall accuracy. The DWT4 is chosen to mitigate risk that the DWT5 may be too computationally intensive for the real-time application by using the same classifier and feature set with only reducing the wavelet decomposition by one level. The final feature set/classifier, CEP26, is chosen to provide contrast to the discrete wavelet transform feature sets by using a feature set based on traditional domain analysis techniques. The modified MFCC only changes the regular MFCC by extending the frequency range to 20 kHz instead of approximately 7 kHz and has the best accuracy of the domain-inspired feature set/classier combinations. These three feature set/classifier combinations are optimized and validated in detail in the following section.

Table 2 Selected feature sets and classifiers for the detailed investigation.

NAME	FEATURE SET	CLASSIFIER	COMMENTS
DWT5	DWPT (Coiflet2, 5 levels)	Bagged trees	Best overall with 94.2% accuracy
DWT4	DWPT (Coiflet2, 4 levels)	Bagged trees	4th best overall with 93.3% accuracy
CEP26	Modified MFCC	Cubic SVM	7th best overall with 92.8% accuracy

B. Detailed Investigation

The detailed investigation of feature sets and classifier algorithms is limited to the three feature sets and classifiers chosen in Table 2. The hyperparameters of each classifier are optimized to find the best performance for each pair and then the models are rigorously tested with an expended set of performance metrics to better understand the expected

real-world performance. Again, all training and hyperparameter optimization is done holding the signal-to-noise ratio of the contaminated blocks to 6 dB, whereas the rigorous testing will vary the signal-to-noise for new, randomized simulated signals. The performance metrics recorded are the accuracy, the false positive rate (FPR), the false negative rate (FNR), and the F1 score. The false positive rate is the proportion of falsely identified contaminated signal blocks out of the total number of known true clean signal blocks. The false negative rate is the proportion of the falsely identified clean signal blocks out of the total number of known true contaminated signal blocks. The F1 score is the harmonic mean of precision and recall, and similar to the accuracy but sensitive to skewness between the false positive rate and the false negative rate [24]. The F1 score is computed by

$$F1 = 2\left(\frac{1}{P} + \frac{1}{R}\right)^{-1},$$

where P is precision (the rate at which the predicted contaminated blocks are actually contaminated) and R is recall (the rate at which true contaminated signal blocks are predicted as contaminated and is also known as the true positive rate).

After each classifier is optimized with respect to its hyperparameters, it is tested through simulation to gauge its robustness for randomization of contamination and then systematically studied for its performance as the signal-to-noise ratio and proportion of signal block is varied. All random draws are made from a uniform distribution for the given parameter and parameter range. The study of the randomization of contamination is performed via a Monte Carlo simulation where a random proportion of a clean signal of length 2 seconds is chosen. Then the clean signal block is chosen to be contaminated or not, and if so, contaminated by a random 2 second block of a randomly selected contamination signal. The classification features are then generated, and a class prediction made using the trained model. This is repeated for a total of 10,000 iterations with all sampling done with replacement and the performance metrics computed. The robustness to the signal-to-noise ratio and the proportion of the signal block contaminated is performed by repeating the same Monte Carlo simulation multiple times varying the value of the either the signal-to-noise ratio or the proportion of the signal block contaminated. The signal-to-ratio is varied from 6 dB to 18 dB in steps of 3 dB. The proportion of the signal block contaminated is varied from 25% to 100% in steps of 25%. A final Monte Carlo simulation is performed where all signal blocks are randomly drawn from the audio recording and randomly contaminated with random values for signal-to-noise ratio and the proportion of the signal block contaminated. This final simulation is carried out for 100,000 iterations.

1. DWT5

The bagged trees model hyperparameters are optimized using a Bayesian optimization approach [29] using 60 iterations training with the discrete wavelet transform feature set using the Coiflet2 wavelet with 5 level decomposition. The improvement in the cross-validation accuracy is seen to converge to within 1% within the 60 iterations. The hyperparameters and final optimized values are:

- Number of individual decision trees: 400
- Maximum number of splits per a decision tree: 350
- Minimum leaf size (minimum number of observations per leaf node of the tree): 1
- Number of predictor variables to sample for each individual decision tree: 35
- Split criterion for splitting a node into two branches: maximum deviance reduction

A summary of the performance metrics for the optimized DWT5 classifiers are show in Table 3. The optimized bagged tree model achieves a cross-validation accuracy of 95.3%, an improvement of 1.1% as compared to the default values used in the general survey of features and methods in the previous section for the same feature set and model.

Table 3 Performance of the DWT5 classifier from the training the optimized hyperparameters, from the randomized contamination simulation, and from the full 100,000 iteration Monte Carlo simulations with all parameter varied.

DWT5	F1 SCORE	ACCURACY	FNR	FPR
OPTIMIZED	95.2%	95.3%	5.9%	3.6%
RANDOMIZED	93.4%	93.6%	10.4%	2.4%
MONTE CARLO	81.6%	84.1%	29.4%	2.5%

Next, randomized contamination simulation is performed with the results shown in Table 3. The F1 score has decreased to 93.4%, a loss of 1.8% and the accuracy has decrease to 93.6%, loss of 1.7%. The parameter sweep simulations for the signal-to-noise ratio and the proportion of the signal block contaminated are shown in Fig. 6 and Fig. 7, respectively. Fig. 6 shows that the F1 score and accuracy decrease with increasing signal-to-noise ratio and that is contributed by the increase in the false negative rate. This means that the classifier is failing to identify

contaminated signal blocks when the signal-to-noise ratio increases. Practically, the contaminated signal is indistinguishable from the clean signal block at approximately a signal-to-noise ratio of 10 dB, where the power contained in the clean signal is an order of magnitude greater than the contamination. The variation in the performance metrics from the variation in the proportion of the signal block contaminated is secondary to the signal-to-noise ratio where Fig. 7 shows the F1 score varies between 93.6% for 100% contaminated to 87.4% for 25% contaminated.

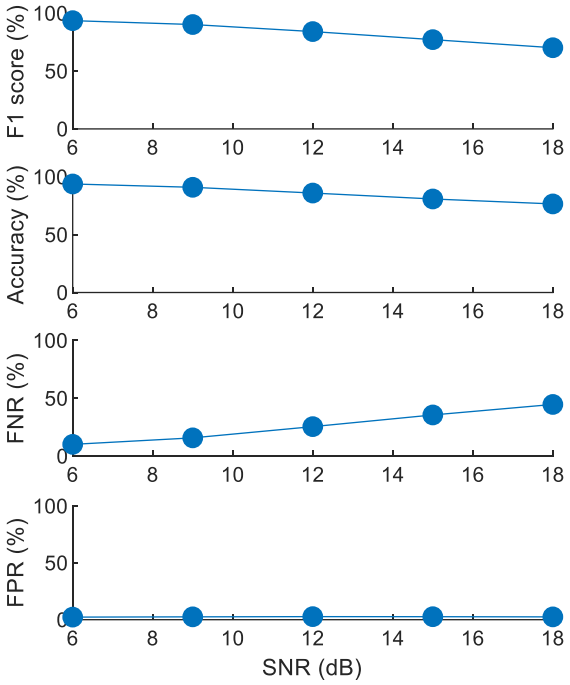

Fig. 6 Performance of the DWT5 classifier with varying signal-to-noise ratio.

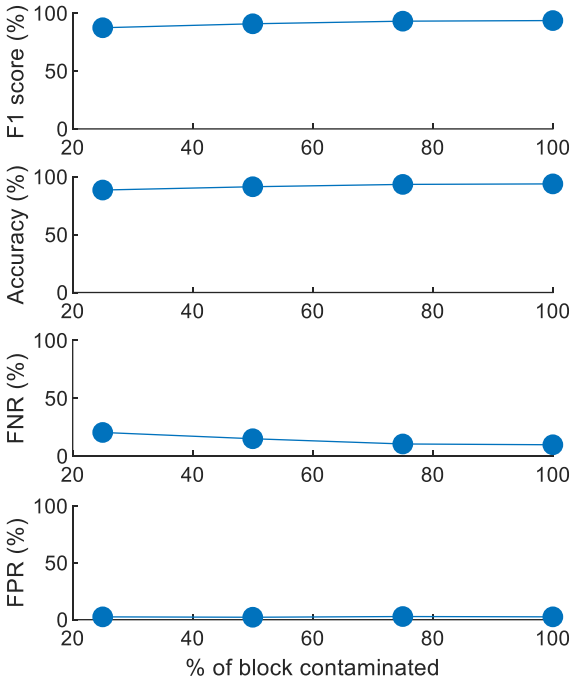

Fig. 7 Performance of the DWT5 classifier with varying proportion of signal block contaminated.

The results for the final Monte Carlo simulation with varying all parameters is shown in the last row of Table 3. The performance is significantly reduced as compared to the cross-validation performance of training the optimized classifier and the simulation with just the randomized contamination blocks. The results of the Monte Carlo simulation confirm that approximately 50% of the 100,000 iterations are contaminated such that there is an even split in the data with regards to the two classes. Again, the false negative rate is significantly increased as there is variation in the signal-to-noise ratio that increases the error of the classifier for misclassifying contaminated signal blocks as clean.

2. DWT4

The bagged trees model hyperparameters are optimized using the same Bayesian optimization approach [29] using 60 iterations training with the discrete wavelet transform feature set using the Coiflet2 wavelet with 4 level decomposition. The improvement in the cross-validation accuracy is seen to converge to within 1% within the 60 iterations. The hyperparameters and final optimized values are:

- Number of individual decision trees: 203
- Maximum number of splits per a decision tree: 2,435
- Minimum leaf size (minimum number of observations per leaf node of the tree): 2
- Number of predictor variables to sample for each individual decision tree: 29
- Split criterion for splitting a node into two branches: maximum deviance reduction

A summary of the performance metrics for the optimized DWT4 classifiers are show in Table 4. The optimized bagged tree model achieves a cross-validation accuracy of 94.8%, an improvement of 1.5% as compared to the default values used in the general survey of features and methods in the previous section for the same feature set and model.

Next, randomized contamination simulation is performed with the results shown in Table 4. The F1 score has decreased to 93.6%, a loss of 1.2% and the accuracy has decrease to 93.8%, loss of 1.0%. The parameter sweep simulations for the signal-to-noise ratio and the proportion of the signal block contaminated are shown in Fig. 8 and Fig. 9, respectively. Fig. 8 shows that the F1 score and accuracy decrease with increasing signal-to-noise ratio and

that is contributed by the increase in the false negative rate. This means that the classifier is failing to identify contaminated signal blocks when the signal-to-noise ratio increases. Practically, the contaminated signal is indistinguishable from the clean signal block at approximately a signal-to-noise ratio of 10 dB, where the power contained in the clean signal is an order of magnitude greater than the contamination. The variation in the performance metrics from the variation in the proportion of the signal block contaminated is secondary to the signal-to-noise ratio where Fig. 9 shows the F1 score varies between 93.6% for 100% contaminated to 87.5% for 25% contaminated.

Table 4 Performance of the DWT4 classifier from the training the optimized hyperparameters, from the randomized contamination simulation, and from the full 100,000 iteration Monte Carlo simulations with all parameter varied.

DWT4	F1 SCORE	ACCURACY	FNR	FPR
OPTIMIZED	94.8%	94.8%	5.7%	4.7%
RANDOMIZED	93.6%	93.8%	10.1%	2.2%
MONTE CARLO	81.4%	83.8%	29.9%	2.2%

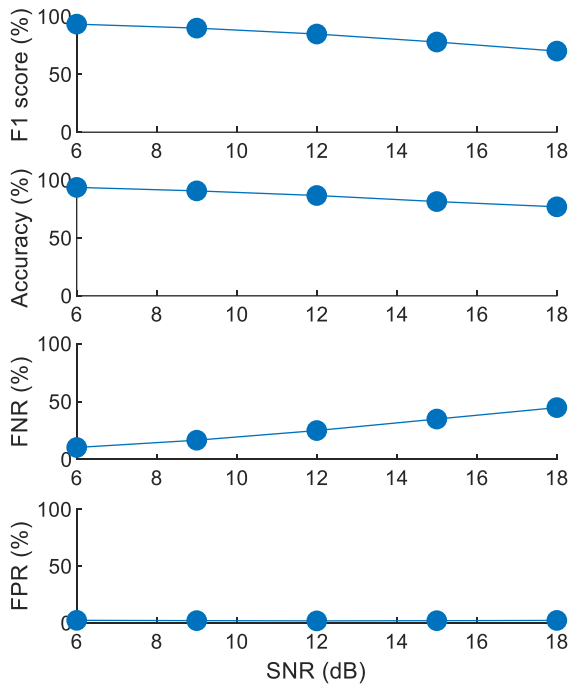

Fig. 8 Performance of the DWT4 classifier with varying signal-to-noise ratio.

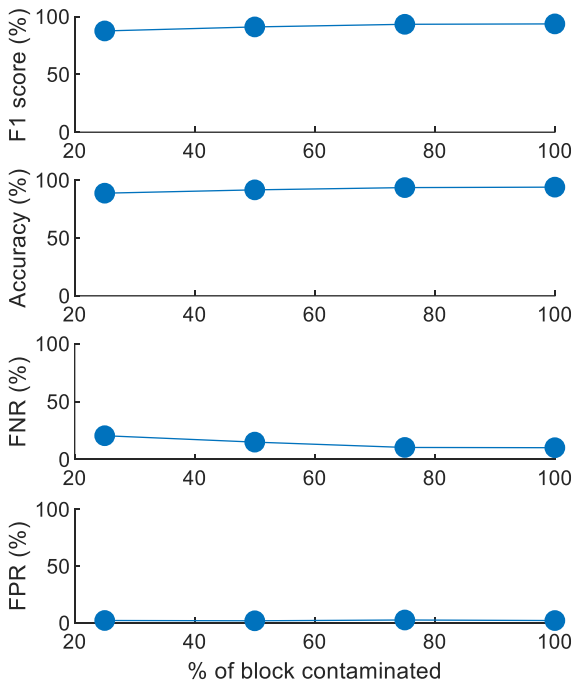

Fig. 9 Performance of the DWT4 classifier with varying proportion of signal block contaminated.

The results for the final Monte Carlo simulation with varying all parameters is shown in the last row of Table 4. Again, the performance is significantly reduced as compared to the cross-validation performance of training the optimized classifier and the simulation with just the randomized contamination blocks. The results of the Monte Carlo simulation confirm that approximately 50% of the 100,000 iterations are contaminated such that there is an even split in the data with regards to the two classes. Again, the false negative rate is significantly increased as there is variation in the signal-to-noise ratio that increases the error of the classifier for misclassifying contaminated signal blocks as clean.

3. CEP26

The cubic SVM model hyperparameters are optimized using the same Bayesian optimization approach as used to identify the ideal parameters for the DWT5 model. The improvement in the cross-validation accuracy converged quickly to nearly the minimum observed objective value. Further iterations only marginally improve it. The hyperparameters and final optimized values are:

- Box Constraint: 0.31037

- Kernel Scale: 2.4063

A summary of the performance metrics for the optimized CEP26 classifiers are show in Table 5. The optimized cubic SVM model achieves a cross-validation accuracy of 93.1%, an improvement of only 0.3% as compared to the default values used in the general survey of features and methods in the previous section for the same feature set and model.

Table 5 Performance of the CEP26 classifier from the training the optimized hyperparameters, from the randomized contamination simulation, and from the full 100,000 iteration Monte Carlo simulations with all parameter varied.

CEP26	F1 SCORE	ACCURACY	FNR	FPR
OPTIMIZED	93.0%	93.1%	8.3%	5.6%
RANDOMIZED	93.6%	93.9%	10.3%	1.9%
MONTE CARLO	83.9%	85.9%	26.4%	1.8%

As with each of the prior classifiers, a simulation is performed to randomize the location of contamination. The results in Table 5 show that F1 score has increased to 93.6%, a gain of 0.6% and the accuracy has increased to 93.9%, gain of 0.8%. Plots for the parameter sweeps for signal-to-noise ratio and the proportion of the signal block contaminated simulations are shown in Fig. 10 and Fig. 11. As the signal-to-noise ratio increases, the F1 score and accuracy decrease while the false negative rate increases. Fig. 10 highlights this trend. Similar to the performance of DWT5 and DWT4, the loss of accuracy is tied to the failure to identify blocks that contain a contaminated signal. When varying the proportion of the signal block that is contaminated, Fig. 11 shows a decrease in the F1 score and accuracy as the amount of contamination decreases. The F1 score ranges from 93.8% when 100% of the block is contaminated down to 86.9% when 25% of the block is contaminated.

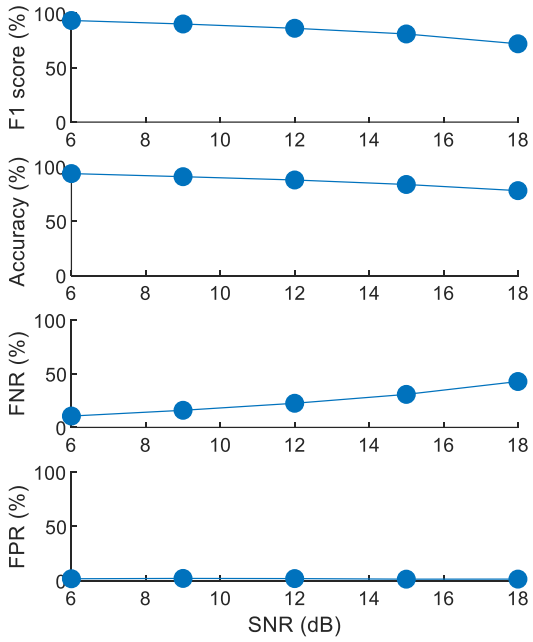

Fig. 10 Performance of the CEP26 classifier with varying signal-to-noise ratio.

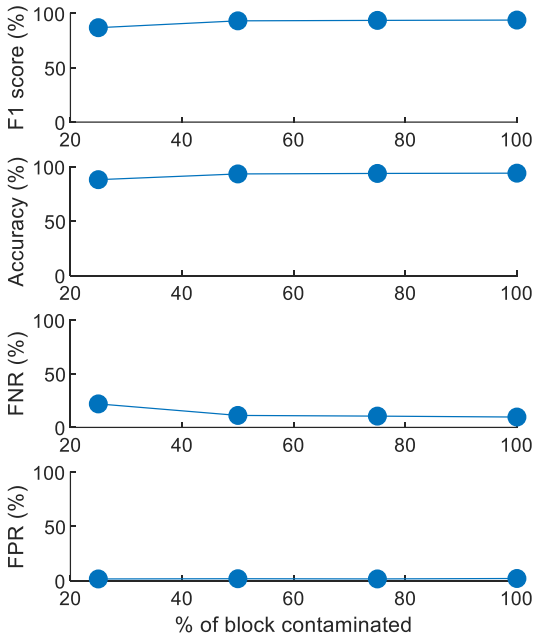

Fig. 11 Performance of the CEP26 classifier with varying proportion of signal block contaminated.

The results for the Monte Carlo simulation where all parameters are varied is shown in the last row of Table 5. The performance in this case is much worse than either the cross-validation performance of the optimized classifier or the simulation with randomized contamination blocks. From the analysis of the results, we can see that about 50% of the iterations were contaminated and there is a uniform distribution in the signal-to-noise ratio. Again, the decrease in accuracy and F1 score are tied to the increased rate at which the model fails to identify contaminated signal blocks.

C. Results Summary

The results for the randomized contaminated signal blocks simulations for each feature set and classifier pair for the detailed study are shown in Table 6. All three combinations performed similarly to each other within a 0.3% for the F1 score and the accuracy. The false positive rate for all is less than 3%, however the false negative rate for all three classifiers is great than 10%. Thus, the classifiers are more likely to miss identifying a contaminated signal block than they are to reject a clean signal block. This could have testing benefits in that the risks of having to redo acoustic measurements in the experiment are lessened, but at the expense of misidentification. However, this risk can be mitigated for continuous contamination sources by the continuous processing and classifying of multiple, overlapping signal blocks and taking a majority vote.

Table 6 Performance summary of optimized feature set and classifier pairs for detailed study subjected to the randomized signal blocks contaminated.

	F1 SCORE	ACCURACY	FNR	FPR
DWT5	93.4%	93.6%	10.4%	2.4%
DWT4	93.6%	93.8%	10.1%	2.2%
CEP26	93.6%	93.9%	10.3%	1.9%

V. Conclusion

An automated process is investigated for classifying aircraft noise signatures with and without environmental noise contamination relevant for community noise fly-over testing. A variety of acoustic feature sets and machine learning classifiers are tested against simulated data derived from real measured audio recordings of aircraft fly-over noise and environmental contamination such as wildlife vocalizations. The investigation is separated into two parts with the first part focusing on breadth and a variety of methods and the second part focused on detailed optimization of three selected combinations of a feature set and classifier.

The results from the broad investigation phase demonstrated the ability of the proposed automated processing to identify contaminated signal blocks with an accuracy exceeding 90% via cross-validation. In particular, traditional machine learning algorithms such as support vector machines and ensemble bagged trees classifiers are performing better than deep learning methods such as convolutional neural networks and are much less computationally expensive. However, the performance of the convolutional neural networks and the LSTM neural networks are lower than expected given their prevalence in machine learning literature. The convolutional neural network showed potential with an accuracy of 92.9% but required a specialized graphical processing unit (GPU) computing card and hours of computational time to train. The traditional and ensemble classifiers are able to be trained within minutes of computational time on an ordinary desktop central processing unit (CPU). Thus, the execution time increase is not justified by the achieved accuracy. Increases in the performance may be achieved by creating feature sets exclusively for the deep learning classifiers such as CNN and LSTM NN and could be a subject for future study.

The results from the detailed study showed that by optimizing the hyperparameters for each of the classifiers that the accuracy of the three chosen combinations are improved by 0.3% to 1.1%. The performance of the classifiers decreased when tested against simulated signal blocks that randomly drawn from the example aircraft audio and randomly drawn from the contamination audio, however, the accuracy still exceeds 90%. The degradation of the performance is increasingly pronounced with a full Monte Carlo simulation creating simulated signal blocks randomly perturbing the signal-to-noise ratio and proportion of the signal block contaminated. The full Monte Carlo simulation resulted in a loss of approximately 10% in the accuracy and the F1 score. This decrease in accuracy can be tied to many cases in the Monte Carlo simulation where the signal block is contaminated but the signal-to-noise ratio is high enough that the signal is indistinguishable from a clean signal and therefore will push the performance metrics lower.

Overall, the proposed algorithms perform well and demonstrate the feasibility of an automated system for environment noise contamination detection. Further work is required to transition the proposed system from this feasibility investigation to a practical system. The authors recommend only continuing with the DWT4 and CEP26 feature set and classifier combinations as they produce similar performance results with less computational requirements than the DWT5 algorithm. In particular, the CEP26 algorithm appears to be the best overall in terms of accuracy and required computation time. The data set should be expanded to include more examples of aircraft noise and an increase in the variety of the examples of noise contamination. This will improve the variety of simulated signal blocks that can be created and used to train the system, which in turn should help increase the generalizability of the classifiers. The simulations should also be improved to include variation in the signal-to-noise ratio from 6 dB to 10 dB where as a signal-to-noise ratio greater than 10 dB is enough such that the contamination is unlikely to impact the overall noise measurements required by the regulators such as the FAA. A methodology should be derived to express

the variation of the proportion of the signal block contaminated as an effective reduction in the signal-to-noise ratio. This may lead to the performance metrics for the variation in proportion of the signal block contaminated into the same curves as for the variation in the signal-to-noise ratio. Finally, the algorithms should be reformatted and investigated for computational performance when used in a streaming, real-time scenario as required by the desired application. The computational performance could limit the deployment of the algorithms if expensive computer resources are required as compared to only inexpensive, single board, system on a chip computer such as the Raspberry Pi or the BeagleBone Black.

Acknowledgments

The authors thank the Boeing Test & Evaluation unit of The Boeing for providing the recorded aircraft audio data and Dr. Megan Hazen for project guidance.

VI.References

[1] Code of Federal Regulations, *Title 14, Part 36, Noise Standards: Aircraft Type and Airworthiness Certification,* Washington, D.C.: Federal Aviation Adminstration, 2016.

[2] International Civil Aviation Organization, *Standards and Reommended Practices, Annex 16, Vol. 1-Aircraft Noise*, 7th ed., Montreal: International Civil Aviation Organization, 2014.

[3] A. V. Dam, "Using the best data possible, we set out to find the middle of nowhere," 20 Feb 2018. [Online]. Available: https://www.washingtonpost.com/news/wonk/wp/2018/02/20/using-the-best-data-possible-we-set-out-to-find-the-middle-of-nowhere. [Accessed 20 Jan 2019].

[4] US National Parks Service, "Sounds Gallery," [Online]. Available: https://www.nps.gov/subjects/sound/gallery.htm. [Accessed 12 Feb 2018].

[5] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature,* vol. 521, pp. 436-444, 2015.

[6] S. Dieleman and B. Schrauwen, "End-to-end learning for music audio," in *IEEE International Conf. Acoustics, Speech and Singal Processing*, Florence, Italy, 2014.

[7] J. Andén and S. Mallat, "Deep scattering spectrum," *IEEE Trans Signal Processing,* vol. 62, no. 16, pp. 4114-4128, 2014.

[8] S. Adavanne, G. Parascandolo, P. Pertilä, T. Heittola and T. Virtanen, "Sound event detection in multishannel audio using spatial and harmonic features," in *Detection and Classification of Acoustic Scenes and Events*, Budapest, Hungary, 2016.

[9] P. Zhang, "Model selection via multifold cross validation," *The Annals of Statistics,* vol. 21, no. 1, pp. 299-313, 1993.

[10] L. R. Rabinar and R. W. Schafer, Theory and Applications of Digital Speech Processing, Upper Saddle River, NJ: Pearson, 2010.

[11] ""ETSI ES 201 108 V1.1.3 (2003–09)", ETSI Standard Document Speech Processing Transmission and Quality Aspects (STQ); Distributed Speech Recognition; Front-End Feature Extraction Algorithm; Compression Algorithms.".

[12] "Classify Time Series Using Wavelet Analysis and Deep Learning - MATLAB & Simulink Example," The MathWorks, Inc, 2019. [Online]. Available: https://www.mathworks.com/help/wavelet/examples/signal-classification-with-wavelet-analysis-and-convolutional-neural-networks.html. [Accessed 12 Jan 2019].

[13] "Image Category Classification Using Bag of Features - MATLAB & Simulink," The MathWorks, Inc, 2019. [Online]. Available: https://www.mathworks.com/help/vision/examples/image-category-classification-using-bag-of-features.html. [Accessed 2019 Feb 7].

[14] J. O. S. III, "Spectral Audio Signal Processing," Center for Computer Research in Music and Acoustics (CCRMA) Stanford University , 2011. [Online]. Available: http://ccrma.stanford.edu/~jos/sasp/. [Accessed 2019].

[15] A. N. Akansu, "Wavelets and filter banks. A signal processing perspective," *IEEE Circuits and Devices Magazine,* vol. 10, no. 6, pp. 14-18, 1994.

[16] A. Graps, "An Introduction to Wavelets," *IEEE Computational Science & Engineering*, vol. 2, no. 2, pp. 50-61, 1995.

[17] N. A. Akansu, I. Selesnick and W. A. Serdijn, "Wavelet Transforms in Signal Processing: A Review of Emerging Applications," *Physical Communication,* vol. 3, no. 1, pp. 1-18, 2010.

[18] The Mathworks Inc, "Matlab Examples: Signal Classification Using Wavelet-Based Features and Support Vector Machines," [Online]. Available: https://www.mathworks.com/examples/wavelet/mw/wavelet-ex52408711-signal-classification-using-wavelet-based-features-and-support-vector-machines. [Accessed 2019].

[19] T. Li and M. Zhou, "ECG Classification Using Wavelet Packet Entropy and Random Forests," *Entropy*, vol. 18, no. 8, p. 285, 2016.

[20] D. P. Percival, "On Estimation of the Wavelet Variance," *Biometrika,* vol. 82, no. 3, p. 619, 1995.

[21] G. Tzanetakis, G. Essl and P. R. Cook, "Audio Analysis using the Discrete Wavelet Transform," in *WSES International Conference Acoustics and Music: Theory and Applications* , 2001.

[22] G. James, D. Witten, T. Hastie and R. Tibshirani, An Introduction to Statistical Learning: with Applications in R, New York: Springer, 2013.

[23] T. Hastie, R. Tibshirani and J. Friedman, The Elements of Statistical Learning, 2nd ed., New York: Springer, 2009.

[24] N. Chinchor, "Evaluation metrics," in *Proc. 4th Message Understanding Conf.*, 1992.

[25] M. T. Hagan, H. B. Denmuth and M. H. Beale, Neural Network Design, Boston, MA: PWS Publishing, 1996.

[26] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks.," *arXiv:1511.08458 [cs.NE]*, 2015.

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Angeulov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions.," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1-9, 2015.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[29] P. I. Frazier, "A tutorial on bayesian optimization," *arXiv preprint arXiv:1807.02811,* 2018.