AFRL-RI-RS-TR-2021-062

INFRASONIC CYBER-PHYSICAL MASINT FOR ENVIRONMENTAL CHARACTERISTICS AND CLASSIFICATION

FLORIDA INSTITUTE OF TECHNOLOGY

APRIL 2021

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2021-062 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /	**/ S /**
JEFFREY T. CARLO	JAMES S. PERRETTA
Work Unit Manager	Deputy Chief, Information Exploitation & Operations Division Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* APRIL 2021	**2. REPORT TYPE** FINAL TECHNICAL REPORT	**3. DATES COVERED** *(From - To)* SEP 2018 – AUG 2020

4. TITLE AND SUBTITLE

INFRASONIC CYBER-PHYSICAL MASINT FOR ENVIRONMENTAL CHARACTERISTICS AND CLASSIFICATION

5a. CONTRACT NUMBER
FA8750-18-2-0113

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
62303E

6. AUTHOR(S)

Anthony O. Smith, Adrian Peter - Florida Institute of Technology
Milton A. Garcés -- RedVox, Inc.
Anand Rangarajan -- University of Florida

5d. PROJECT NUMBER
FIT1

5e. TASK NUMBER
80

5f. WORK UNIT NUMBER
01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

PRIME
Florida Institute of Technology - 150 W University Blvd, Melbourne FL 32901

SUB
RedVox, Inc. -- PO Box 1599, Kailua Kona, HI 96745
University of Florida -- PO Box 113001, Gainesville, FL 32611

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RIGC
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER
AFRL-RI-RS-TR-2021-062

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report was developed under a Cooperative Agreement. This work developed an advanced machine learning (ML) technique that is uniquely capable of exploiting heterogeneous and potentially lower-fidelity, cyber-physical signatures. In this effort, we moved beyond traditional collection and analysis of these signatures, instead utilized mobile devices to capture infrasonic signatures. These distributed mobile devices can be used to collect low frequency sound waves from targets of interest, such as small to medium UAVs, artillery, and other targets of interest. Report presents classification results for some targets of interest.

15. SUBJECT TERMS

Acoustic, Machine Learning, Time series, infrasound, wavelets, edge analytics, IoT

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** **JEFFREY T. CARLO**
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UU	43	**19b. TELEPHONE NUMBER (*Include area code*)** **N/A**

Table of Contents

List of Figures and Tables

List of Tables

List of Equations

1 Summary

The report herein details the overall accomplishments and lessons learned of the *Infrasonic Cyber-Physical MASINT for Environmental Characterization and Classification*. The project explored Artificial Intelligence/Machine Learning (AI/L) techniques to classify acoustic signals, independently from collected or other relevant datasets. Formulated as a feature learning and discrimination problem, we draw comparative conclusions between techniques that consider unique features for classification. Finally, we offer recommendations based on the techniques' strengths and weaknesses and the approaches taken to arrive at our conclusions.

Consider that land and air vehicles (e.g., tanks, trucks, helicopters, and UAVs) exhibit acoustic power spectrum from several hundredths of one Hertz (Hz) to a few hundred Hz. Acoustic sources with impulsive characteristics such as artillery and UAVs have broadband spectral energy, often overlapping ranges. With these targets of interest generating spectral content in harmonious frequency bands, acoustic sensors are concomitantly suited for event classification and identification.

At the heart of Measurement and Signatures Intelligence Exploitation (MASINT-X) is the desire to locate, track, identify, or characterize sensor observations of high-value targets and surroundings. Ideally, signatures exhibiting a high signal-to-noise ratio (SNR) are collected via high-fidelity sensors. Observations align with reliable physical models of how the waveforms are generated and propagated in the environment. These ideal conditions are rarely met, and system specifications (and associated cost) can substantially exceed collected signal quality. Although the traditional MASINT process has yielded invaluable intelligence, the mission planning and resource allocation needed for these larger sensor platforms often introduce untimely delays in the race to predict and prevent adversarial tactics. The emerging sub-discipline of cyber-physical MASINT adopts a non-traditional approach to signal intelligence. It seeks to exploit the cyber domain-the realm of connected devices-to obtain signatures from the slew of sensors typically integrated into modern-day technological devices. However, what we gain in terms of immediate accessibility and crowd-sourced intelligence can be degraded by lower sensor sensitivities and unpredictable system responses that cause high levels of uncertainty in event detection, identification, and characterization.

2 Introduction

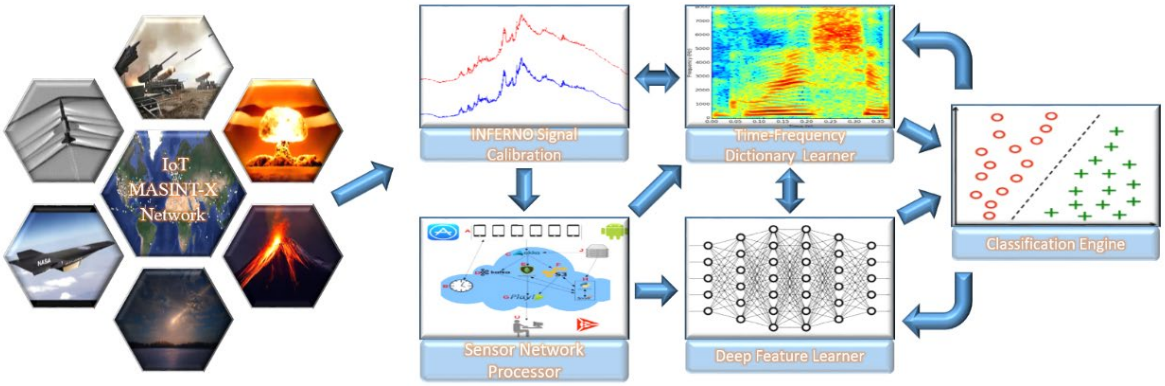


Figure 1 Deep Discriminative Dictionaries for Infrasound Classification (D3IC).

This work aims to develop *advanced machine learning (ML) techniques* that are uniquely capable of *exploiting heterogeneous and potentially lower-fidelity, cyber-physical signatures*. Our highly qualified team is composed of RedVox, Inc.'s for signal collection via a *mobile device*, and both the Florida Institute of Technology, FIT ([1]–[7]), and the University of Florida, UoF ([8]–[10]) for algorithm development.

Figure 1, Deep Discriminative Dictionaries for Infrasound Classification (D3IC) system provides an end-to-end cyber-physical infrasonic processing framework: starting from infrasound signal acquisition through mobile devices (Android and iOS), calibration and network processing for localization and SNR improvement, a novel time-frequency dictionary learning method denoising and compressed representations, feature extraction through deep learning, and an integrated discrimination engine that is feedback to the dictionary and deep learners for improved classification in the presence of low fidelity cyber-physical signals.

Acoustic-based analyses possess many desirable properties, such as long-distance wave propagation and attributable spectral characteristics. However, for the past 20 plus years, it has been, by large, relegated to capturing massive explosions, earthquakes, bolides, and space-bound rocket launches. These collections have been captured by traditional sensor networks, like the International Monitoring Systems (IMS). In this effort, we move beyond traditional collection and analysis of these signatures, instead proposing mobile devices to capture infrasonic signatures. These distributed mobile devices can collect the previously mentioned sources that can be used to classify localized sources, small to medium UAVs, artillery, and other related tactical threats. Also, mobile device infrasonic observations can be correlated (even fused under certain scenarios) with the traditional collections for unprecedented signature analysis. However, what we gain in ubiquitous monitoring, is not without challenges like signal quality.

Traditional sensor array data exhibits more uniformity in the waveform amplitudes and spectral characteristics. This readily allows the data to be processed by traditional algorithmic techniques to improve SNR, perform detection, characterize physical phenomenology, and provide event classification via standard ML approaches. On the other hand, the mobile sensors' data exhibit many irregularities due to effects such as diffraction, dispersion scattering, and non-linear shocks that are compounded in the cyber-physical domain (as expected since the multipurpose device is not optimized to capture infrasound). The same event observed in the traditional domain will appear considerably different from a cyber-physical device. Difficult problems such as these are ideally suited for the algorithmic approaches grounded in machine learning, whose primary purpose is to extract latent patterns and distinguish attributes that often appear obfuscated. Our novel *deep discriminative dictionaries for infrasound classification* (*D3IC*) system employs ML in service of signal processing to yield an integrated system supporting the following state-of-the-art capabilities:

- Developed a cross-platform calibration for mobile device infrasound captures.
- Demonstrated single device, array, and distributed network signal processing localization and SNR enhancement.
- We created time-frequency dictionary learning for denoising and discriminating compressed signal representation.
- We researched deep learning for robust feature extraction from infrasonic cyber-physical signatures.
- We designed a modular ML framework with the flexibility to train and deploy algorithms independently or integrate them for richer classification power.
- We have validated a robust and repeatable ML model development, deployment, and validation process.

Figure 1 illustrates the D3IC system and its major subcomponents that operate in concert to provide these functionalities. Given that we acquire acoustic signals from various mobile devices, it is critically important that the signals are appropriately normalized before further processing the D3IC system. This requirement is addressed by the INFERNO subsystem, which employs a Gabor octave filter to standardize the signals. This approach was pioneered by team member Garcés (RedVox CEO), who has demonstrated its success on traditional sensor arrays [1], and more recently, on Android and iOS devices. The Sensor Network Processor (SNP) contains a suite of array and distributed network signal processing algorithms for event localization and SNR improvement [e.g., F-K analysis, subspace methods, network processing ([2]–[5])]. If correlations cannot be established across sensors, the SNP serves as a pass-through so that signals can be processed individually. The Time-Frequency Dictionary Learner (TFDL) and Deep Feature Learner (DFL) subsystems both employ (similar-in-spirit) hierarchical network structures to produce highly compressed sparse representations and rich feature descriptors. The TFDL utilizes well-known Gabor wavelet frontends and proceeds with learning denoised representations using discriminative dictionaries. The DFL performs feature extraction using an efficient deep learning pipeline that employs convolutional and standard neural layers. The processing chain culminates in the Classification Engine (CE), where we use a support vector machine (or equivalent) for state-of-the-art classification. We have significantly enhanced the utility of these three methods by mathematically formalizing a loss objective that connects the TFDL and DFL through feedback from the CE. This tight integration is optional; the flexibility exists to evaluate the effectiveness of coupling or decoupling these components empirically.

As constructed, D3IC is a modular system that supports various functional threads on how the subsystems are interconnected to form an end-to-end processing chain. For example, it is possible to employ only INFERNO, SNP, TFDL, and the CE to produce an event classification framework. Later in the development cycle, DFL can be substituted for TFDL, or even used in conjunction, as alluded to earlier. From our extensive ML experience, we have found that such a "plug-n-play approach" is ideally suited for ML models, which are constantly updated and often evaluated among a set of alternatives. The modular approach also enables us to define interfaces easily and begin rapid prototyping without too many cumbersome interdependencies.

3 Tasks

Task #1: **Program Management and Oversight**

The FIT was the primary for the management and technical oversight of the efforts described in this technical report.

Task #2: Data Collection, Standardization, and Dissemination

RedVox, in collaboration with FIT, worked to identify targets of interest and perform a controlled collection of infrasound signals with a supported mobile device.

- **Subtask**: Identify customer targets of interest.
- **Subtask**: Define collection requirements (equipment, location, etc.)
- **Subtask**: Collect multiple target infrasound signals on iOS and Android platform devices.
- **Subtask:** Signal processing for calibration and standardization.
- **Subtask**: Establish secure, controlled access and storage of collected data samples.
- **Subtask:** Expert analysis for signal collection and dissemination.
- **Subtask:** Maintain and update RedVox app and cloud server.

Deliverable(s): Collection of signals of interest. Cloud-based app and environment for data collection.

Task #3: Signal denoising and wavelet dictionary learning

The UoF, in collaboration with FIT, investigated and developed processing techniques for signal enhancement(s). We accomplished infrasound data signal processing using a set of time-frequency-scale Gabor filter wavelets, which together represent the infrasound signal.

- **Subtask**: Gabor-wavelet signal representation.
- **Subtask**: Examine and identify a set of scales and frequencies to match the infrasound domain.
- **Subtask**: Denoise and discriminative dictionary learning. Gabor coefficients extracted from the signal using over-complete dictionary representation.
- **Subtask**: Explore standard methods such as K-SVD and LASSO for sparse signal representation in an over-complete basis.

Deliverable(s): Time-frequency dictionary for denoising and discriminatively compressed infrasound signal representations.

Task #4: Analysis of robust feature extraction

In collaboration with the UoF, FIT employed techniques for robust feature extraction from infrasonic cyber-physical signatures that will enable class discrimination.

- **Subtask**: Analysis of Gabor coefficients as ideal features.
- **Subtask**: Analysis of dictionary projections as robust features.
- **Subtask**: Explore deep convolutional networks for feature extraction.

Deliverables: A robust feature set, ideal for signal class discrimination.

Task #5: Machine learning and Deep Networks for classification

In collaboration with the UoF, FIT performed a classification of infrasound signals utilizing a standalone Support Vector Machine (SVM), deep neural network, or a mixed classification approach.

- **Subtask**: Deep network modeling to perform classification.
- **Subtask**: Explore the viability of a majorized support vector machine hinge loss classifier.
- **Subtask**: Mixed classifier approach with deep network and SVM.

Deliverables: Classification model for class discrimination.

Task#6: Integrated prototype capability for processing mobile device infrasound signals

FIT developed a prototype modular ML framework with the flexibility to train and deploy algorithms independently or integrate them for richer classification power. Robust and repeatable ML model development, deployment, and validation process.

- **Subtask**: Design a prototype system for deployment.
- **Subtask**: Develop a prototype system for end-to-end processing.
- **Subtask**: Develop and perform test cases to demonstrate capability.
- **Subtask**: Provide system design documents and technical instructions for operation.

Deliverables: A prototype system was delivered to demonstrate infrasound class discrimination. We employed a modular approach since there are many ways in which the above modules can be configured. Furthermore, at each stage, there is potential for improvement via feedback to an earlier module. After testing and delivering the individual modules, we tested various feedback system configurations incorporating the above modules.

Table 1 Summary of task and deliverables.

Task #	Description	Deliverable
#2	Data collection, standardization, and dissemination.	Collection of signals of interest. Cloud-based app and environment for data collection.
#3	Signal denoising and wavelet dictionary learning.	Time-frequency dictionary learning for denoising and discriminatively compressed signal representation.
#4	Analysis of robust feature extraction.	A robust feature set, ideal for signal class discrimination.
#5	Machine learning and deep networks for classification.	Classification model for class discrimination.
#6	Integrated prototype capability for processing cyber-physical infrasound signals.	A prototype system to demonstrate infrasound class discrimination.

4 Methods, Assumptions, and Procedures

4.1 INFERNO Signal Calibration and Standardization

We integrated the strengths of two platforms: INFERNO (Infrasonic Nth Octave multiresolution energy estimator) software platform for infrasound signal normalization and its close relationships with the Gabor wavelet machine learning frontend.

Infrasound signals can span the spectrum from 0.001Hz to 20Hz. This is a spread of four orders of magnitude and is, therefore, a decade broader than the audio range (typically 20Hz to 20 kHz). While we do not discuss the sensor spread here (since the sensor array processing network is discussed in the next section), their spatial distribution can range from meters to kilometers. Furthermore, the energy spectra of infrasonic signals can vary by twelve orders of magnitude. When the diverse ranges of the spatial, spectral, temporal, and amplitude scales are considered, the challenges facing a unified approach to infrasound processing become more apparent. Finally, the collection process for infrasound signals is fraught with data gaps, noise clipping, distortion, transience, etc. There is, therefore, a clear and present need for infrasound signal normalization—a task addressed by Co-PI Garcés previous work [11] that introduced the INFERNO approach.

The INFERNO platform has mainly been concerned with 24-bit digital pressure waveforms that correspond to pressure waves traveling outward from a source (at the speed of sound) with recorded pressure fluctuations. While these infrasound signals have been popular in remote sensing, only recently has cellphone devices and other 21st century digital collection technologies stepped forefront. This, in turn, has widened the scope of infrasound signal processing, making it more necessary to study signal collection, processing, normalization, denoising, classification, and characterization in this domain. INFERNO provides a one-of-a-kind platform for signal normalization and characterization of energy-based signatures, ideally suited to the present-day twin demands of decentralized data collection and cloud-based signal processing and machine learning.

INFERNO addresses the orders of magnitude scaling of infrasound frequencies by performing fractional octave processing. Frequency octaves are typically divided into 1/3 bands with these fractional octave schemas resulting in overlapping, narrow frequency processing filter banks. Since time-frequency analysis entails corresponding time domain filter representations, the temporal scales are also carefully worked out, with the multiple scales dovetailing one another as expected. Note that this logarithmic frequency processing has a clear counterpart in the Gabor wavelet processing downstream. Therefore there is a clear synergy between the two frameworks, which can be exploited by twin processing of denoising and normalization instead of the more conservative pipelined option. INFERNO performs its own Gabor filter-based processing of signals via its carefully constructed set of filter banks and temporal windows. Amplitude normalization is adorned with basic denoising capabilities since the fractional wavelet bin schema includes signal integration within narrow frequency bins. Further, the signals' spectral energies are also computed (with attempts to separate the noise energy components) – an energy signature-based characteristic of the signals. This last feature has a lot in common with the projections of signals onto Gabor wavelet bases (contemplated as the first stage in denoising), strongly suggestive that normalization, energy signature-based characterization, and denoising should proceed in lockstep.

When considered within the larger landscape of signal processing and machine learning, INFERNO's capabilities suggest a set of deliverables aimed first at prosaic modular components followed by a more ambitious synthesis, especially of normalization and denoising. The initial phase of the project included the development of and execution of INFERNO and the Gabor wavelet processing frontends in pipelined mode. Subsequently, after becoming more familiar with how these algorithms affected our results we then integrated INFERNO and Gabor wavelet denoising at a more fundamental level to benefit both methods.

4.2 Sensor Network Processor

The system is designed to collect and classify signals acquired from multiple mobile devices. When multiple devices are spatially located within a desirable distance, we can then employ techniques from directional arrival (DOA) estimation [12] to provide additional characterization information regarding the location of the source event and possibly enhance the SNR of the captured signals. The Sensor Network Processor (SNP) subsystem is designed to address signal-processing needs associated with array and network processing. Strictly speaking, array processing techniques like digital beamforming [13] require certain physical requirement be satisfied, e.g., spatial separation d between the array elements must be $d \leq \frac{\lambda}{2}$, where λ is the is the wavelength associated with the carrier frequency. Though it is possible to construct such an array with mobile devices, we cannot rely on non-cooperative mobile users to synchronize in the required array geometry. If by chance they do, we can certainly take advantage of such scenarios to invoke coherent processing. However, most of the time, we anticipate users will be within sufficient proximity of an event of interest, given that infrasound wavelengths support propagation for fairly long distances (depending on the source event's strength). The SNP will support several coherent ([14], [15]) and non-coherent ([16], [17]) “array” processing methods. Since the mobile devices provide timestamp and geo-location information for the devices, the infrasound signal capture from various devices can be intelligently fused to provide both event localization information and improve the SNR. If meaningful meta-data is lacking or not provided by the device user, the SNP simply serves as a pass-through to the ML subsystems of our D3IC system.

4.3 Deep Discriminative Dictionaries for Infrasound Classification (D3IC)

Infrasound signals have special characteristics. The frequency range begins at much lower frequencies than encountered in other typical time series applications. The data are much noisier and more diverse than their standard time series counterparts. The paucity of data translates to severely curtailed availability of pre-trained deep learning layers. For these reasons, we cannot expect to make significant progress in infrasound data processing by simple-minded reliance on deep networks operating directly on the original signals.

The onerous denoising and feature extraction requirements in infrasound signal processing prompt us to (a) begin with time-frequency analysis of the signals using short-time Fourier transforms or Gabor wavelets operating at multiple frequencies and scales and (b) incorporate discriminative dictionary learning on the estimated time-frequency-scale parameters to perform denoising, (c) feature extraction simultaneously, and (d) classification. Consequently, we envisage a hierarchical network structure that begins with well-known Gabor wavelet frontends. Then proceeds with learning a denoised representations using discriminative dictionaries. Next, performs feature extraction using an efficient deep learning pipeline that culminates in a support vector machine (or equivalent) state of the art classification. This unique blend of parametric signal processing, non-parametric deep learning, and convex optimization-driven support vector classification is ideal for the infrasound domain.

Time-frequency analysis of signals has a long history ([18], [19]). After the Fast Fourier Transform (FFT) advent, the benefits of applying short time Fourier Transforms (STFT) to signals became clear, especially in non-stationary situations. This line of research continued with wavelets (which incorporated multi-resolution processing) and Gabor wavelets. The combination of scale and short time-frequency analysis afforded by Gabor wavelets make them well suited to this domain [20]. We are using the scales and frequencies discussed in [11] to obtain detailed time-frequency-scale characteristics of infrasound signals. The frequency range of the Gabor wavelets can be tuned to the infrasound range, and the scale parameters can be set based on the signal lengths. While the above description of the scale and time-frequency processing using Gabor wavelets is necessarily (because it is introductory) biased toward signal filtering, we envisage the Gabor wavelets used in dictionary learning mode. That is, we unpack a training set of infrasound signals using a Gabor wavelet dictionary where the "atoms" are individual wavelets centered at time points and focused on one scale and frequency. If a signature is well characterized, such as explosive blasts, it is also possible to follow a similar approach with customized wavelets tuned to the signal of interest as demonstrated in [21].

While dictionary learning is now considerably more standardized with readily available [22] suites of algorithms and software, our needs force us into a more unique and specialized dictionary learning aspect. The first point of departure is the need for dictionary learning in infrasound signal space rather than in the more standard vector space of features. The second point of departure is the need for dictionary learning to be discriminative. In other words, it is imperative that the learned dictionary be maximally influenced by feedback from downstream processing of features. Finally, we seek to learn discriminative dictionaries comprising Gabor wavelet atoms as described above. For all these reasons, the D3IC approach is unique but well suited to the infrasound domain. Below, we give an introductory treatment of the dictionary learning component with the caveat that the payoff depends on feedback from downstream deep learning.

The past decade has seen a lot of development in dictionary learning. The initial work with principal component analysis quickly evolved into learning how to estimate overcomplete dictionaries. The K-SVD algorithm [22] and orthogonal matching pursuits (OMP) [23] etc., are popular algorithms in this space. In the most general setting, dictionary learning results in estimating a set of vectors, which constitute an over-complete basis for a training set. Each vector in the training set has multiple exact representations in the dictionary. The learned dictionaries have several applications - denoising and discrimination being popular choices as well as efficient coding. In the denoising application, noisy incoming vectors are projected on to a set of principal or leading components using variance thresholding. Noisy components correspond to subspaces that occur less frequently in the training set. When discrimination or classification is the target application, the learned dictionaries are used to perform (rudimentary) feature extraction prior to classification. A dictionary is deemed discriminative [24] if the prior dictionary learning is the misclassification error (or convex approximations thereof). In this case, the projection onto the basis vectors leads to better class discrimination (instead of just denoising).

The preceding discussion should be clear that discriminative dictionary learning as standardly conceived lacks a deep learning component for obtaining improved features. To this end, we implemented a coupled discriminative dictionary learning [24] with deep learning ([25], [26]). The hierarchical deep learning stages are interjected in between the dictionary learning and classification steps. In this way, dictionary learning leads to a rudimentary set of features amplified by deep learning prior to classification (using a state-of-the-art classifier like the SVM). Consequently, the dictionary's discriminative aspect does not have to bear the whole burden of feature extraction since the deep learning stages can pick up the slack. This permits the learned dictionary to focus on denoising and the extraction of rudimentary features, thereby avoiding an unnecessary compromise between denoising and discrimination. Or in other words, dictionary learning results in denoising and the extraction of a basic set of features amplified by the deep learning component into composite features more ideally suited for classification. There does not exist much previous work coupling denoising and dictionary learning with the deep learning stack.

The integration of deep learning ([25], [26]) (sans the convolutional layers and the final classifier) with dictionary learning primarily involves a marriage of disparate software stacks: TensorFlow [27] and its relatives give us simple APIs for deep learning which must be integrated with dictionary learning. The result is a backpropagation of the first stage derivatives of deep learning on to the backend of the dictionary learning software stack: the basis coefficients are modulated by downstream changes fed back from the minimization of the misclassification error (or convex approximations thereof) ([28], [29]).

The final aspect of infrasound processing concerns the classifier. Rather than use the simpler classifiers in the deep learning stack, we elect to use the best of breed support vector machine (SVM) classifier in the form of the convex hinge loss function (and its multi-class extensions) [30]. The backstop of the deep learning stages is the SVM, with the hinge loss function approximating the misclassification error. We eschew kernel SVMs since the deep learning stages effectively give us composite features that are well-tuned toward discrimination rendering the "kernel trick" unnecessary.

We have given a brief description of the entire system: from Gabor wavelet preprocessing to dictionary-driven denoising and basic feature extraction, continuing with composite feature extraction via deep learning and culminating with an SVM backend. The result is a deep discriminative dictionary-driven processing and classification system – well-tuned to infrasound signals.

4.4 Time-Frequency Denoising: Gabor Wavelet Dictionary

As previously mentioned, time-frequency analysis using Gabor filters and wavelets has become a popular approach due to many factors. Below, we detail an approach to dictionary learning using Gabor wavelet atoms. This is developed in standalone mode to forestall potential confusion without reference to downstream products such as deep learning layers or SVMs. The advantage of standalone development of Gabor dictionaries is a modular payoff. Furthermore, this aspect of our work is driven entirely by the need for denoising. The goal is to provide a denoising module based on Gabor wavelet dictionary thresholding.

$$\phi(t;\nu,s) \equiv \exp(-\frac{t^2}{2s^2})\exp(2\pi i\nu t) \qquad \text{Equation 1}$$

The Gabor "atoms" used in dictionary learning are denoted as where s is a scale parameter and ν the frequency. This Gabor atom is centered at the origin but can easily be shifted to any temporal location. The central goal of dictionary learning is to represent any infrasound signal $f_i(t), i \in \{1, \ldots, N\}$ via an expansion

$$f_i(t) \approx \sum_{k=1}^{M}\sum_{l=1}^{n} c_{kl}^{(i)}\phi(t - t_k; s_{kl}, \nu_{kl}). \qquad \text{Equation 2}$$

This relationship reflects the fact that multiple (n) scales and frequencies are deployed at every location t_k resulting in a total of Mn atoms used for signal representation. The approximate nature of the above relationship is meant to suggest a least-squares error principle for estimation of the set of unknown coefficients $\left\{c_{kl}^{(i)}\right\}$:

$$E_{\text{Gabor}}\left(\left\{c_{kl}^{(i)}\right\}; \{s_{kl}, \nu_{kl}\}\right) = \int_0^T \left| f_i(t) - \sum_{k=1}^{M}\sum_{l=1}^{n} c_{kl}^{(i)}\phi(t - t_k; s_{kl}, \nu_{kl})\right|^2 dt + \kappa \sum_{ikl}\left|c_{kl}^{(i)}\right| \qquad \text{Equation 3}$$

The above objective function contains an ℓ_1 regularization with regularization parameter κ on the Gabor coefficients (which will necessitate LASSO-based optimization) ([31]–[33]). Other norms can, of course, be considered. In the above least-squares problem, it should be understood that the set of scales and frequencies are given or fixed (by the domain). For example, the infrasound domain will dictate the range of frequencies, and the temporal length of the signals (in seconds) will dictate the range of scales. We have taken some liberties in notation here; the Gabor wavelet is complex, whereas the signal is not, which implies that the complex atoms will be used in conjugate pairs etc. The signals are not usually available at all time points but only at discrete locations, whereas an integral symbol is used. In practice, the integration will be carried out numerically when required (though we reserve the right to evaluate some integrals using infinite extensions of the interval analytically) ([34], [35]). Denoising via wavelet thresholding works by thresholding wavelet coefficients; Gabor wavelets are not an exception despite being non-orthogonal.

In previous work on image denoising [36] (which has been well received so far with over 150 citations in 4 years), we estimated complete HOSVD dictionaries from color image patches, which were first chosen based on a ℓ_2 similarity measure. Here, we let the dictionary learning algorithm implicitly do the same with the coefficients performing a similar role.

A simple and direct way to get to the heart of Gabor dictionary learning is to consider the learning process in a pipelined fashion. Given a set of infrasound signals, we first estimate Gabor coefficients $\left\{c_{kl}^{(i)}\right\}$, using regularized versions of the least-squares objective function above. This allows us to transition from the original set of (continuous-time) signals to a set of time-frequency-scale coefficients. The advantage inherent in this transition is that the coefficient set can now be modeled as living in a vector space, thereby allowing standard dictionary learning to be performed. Therefore, we can model Gabor dictionary learning's overall process as encompassing a two-stage process: first, we estimate Gabor coefficients from the original signals, and next, we estimate (possibly over-complete) dictionaries from the coefficient set. We reserve the right to consider feedback in this process, which would entail that stages 1 and 2 above would be coupled. However, a decoupling allows for a more straightforward exposition.

There are different approaches to dictionary learning to consider ([22], [37]). We explored the well-known K-SVD algorithm, which alternates between dictionary updates and projections onto the dictionary bases. Subsequently, we can revisit these choices – including our *award-winning* previous work in this area ([38], [39]) - based on the empirical results. Since the dictionary is over-complete, the computational problem is a very difficult one; therefore, we initialized the dictionary using principal components. The result is a set of bases that serve as generators of Gabor coefficients and, therefore, infrasound signals.

Denoising can be performed within this framework using basis thresholding. In the simplest approach, PCA-based denoising can be performed. Using over-complete bases for denoising is more complex but now has a reasonable track record over the past decade. Note that, for the sake of exposition, we stick to the pipeline described above. After extraction of Gabor coefficients, dictionary learning is performed. Basis thresholds on the dictionary vectors lead to filtering and elimination of noisy coefficients. The new set of coefficients can be used to obtain new, denoised, infrasound signals. Note, no contact has yet been made to the world of classification.

This process can be continued. Sticking with the pipeline motif, we can perform deep learning on the denoised projections of Gabor coefficient sets on to the dictionary bases (which being over-complete requires the use of LASSO or equivalent). It is natural to ask why dictionary learning could not have been performed directly on the infrasound signals instead of on the Gabor coefficients. Since the signals are 1-D functions, dictionary learning in the space of continuous (and perhaps band-limited) functions is quite difficult (since function norms and inner products have to be used as opposed to finite vector space ones). Suppose the functions are discretized and fed into a dictionary learning procedure. In that case, we encounter a permutation problem: usual versions of dictionary learning are invariant to permutation of the discretized values (provided all function discretization's are permuted in the same manner).

In contrast, invariance to permutation of Gabor coefficients is not a problem since they lack temporal ordering. For these reasons, it is reasonable first to extract Gabor coefficients followed by dictionary learning in coefficient space. (The issue of pipelined versus parallel learning is orthogonal to the issue of using discretized infrasound signal values directly in dictionary learning.) Finally, we could have elected to perform denoising directly in the Gabor coefficient space without bringing in dictionary learning. If denoising were the sole goal, this could conceivably be good enough (from a pragmatic perspective), but since we need to denoise and discriminate, we have elected to combine Gabor coefficient extraction with over-complete dictionary learning since this combination facilitates the joint needs of denoising and discrimination.

$$E_{\mathrm{TFDL}}\left(D,\{\boldsymbol{v}^{(i)}\};\{\boldsymbol{c}^{(i)}\}\right)=\sum_{i}\left\|\boldsymbol{c}^{(i)}-D\boldsymbol{v}^{(i)}\right\|_{2}^{2}+\lambda\sum_{i}\left|v_{j}^{(i)}\right| \qquad \text{Equation 4}$$

The TFDL over-complete dictionary learning objective function is where D is the dictionary containing a set of bases, $\{\boldsymbol{v}^{(i)}\}$ a set of projections onto the bases and $\{\boldsymbol{c}^{(i)}\}$, the set notation for the input Gabor coefficients (on which the dictionary is being constructed). The regularization parameter λ precedes an ℓ_1 norm term on the projections since the over-complete bases can lead to several unnecessary projections (and therefore set to zero). This is the standard LASSO sparsity term [31]. Consequently, we can use standard packages such as LASSO and K-SVD [22] to accomplish dictionary learning on the Gabor coefficients. Once again, despite this being a standalone dictionary learning objective function (in the pipelined setting), we can dovetail it with the previous Gabor coefficient extraction step indicating a dictionary learning-driven regularization on the input coefficients.

The past decade has seen huge improvements in dictionary learning algorithms and attendant software suites ([37], [33]). While K-SVD remains popular (and therefore a first-choice approach), the integration of LASSO (convex algorithms) with over-complete basis pursuit can lead down myriad algorithmic pathways. It is extremely rare to see dictionaries constructed on Gabor coefficients [32] (which will be turned into reals using symmetrically placed complex exponentials), and even more rare to see this performed in the infrasound domain. Consequently, this approach is rich with potential empirical discoveries in denoising (over and beyond simple wavelet thresholding). Once the dictionary has been constructed, denoising commences by examining different projections and lossy reconstructions (which eliminate noise). Once again, note that no contact has been made with the world of classification. At this juncture, we are focused on building dictionaries and obtaining infrasound signal reconstructions which are cleaner versions of the original. We expect to perform many statistical tests on the residuals and pick reconstructions based on residual bias and variance. (We have ample experience in this domain from previous work in image denoising.) Finally, despite eschewing NL-means style approaches in denoising, the dictionary effectively does pull in remote infrasound patches similar to each other. With this description of dictionary-based denoising in place, we incorporate deep learning on the projections next.

4.5 Deep Feature Learning

Thus far, we have introduced Gabor wavelet infrasound representations and time-frequency dictionaries. We now detail the deep learning component whose main task is to amplify the intrinsic differences between the incoming Gabor coefficients and/or dictionary representations using labeled data. The centerpiece of this strategy is how we leveraged software suites and libraries to perform this task and, in addition, perform a circular feedback to earlier stages (like time-frequency dictionary learning). Instead, we removed the top classification layer and replace it with a hinge loss support vector machine (SVM). This change is driven by the fact that the SVM is usually a better classifier than the simpler ones used in deep learning.

A standard two-layer deep learning neural network ([25], [26]) minimizes the following objective function during training with respect to the unknown weights W. Note the presence of the nonlinearity ϕ separating the upper layer from the lower one. The label training information is present in the set $Z = \{z^{(i)}\}$. In standard deep learning, these would be the output labels. We reserve the right to have an output layer corresponding to an SVM connected to the deep learning network:

$$E_{\mathrm{DFL}}\left(W; \{\boldsymbol{z}^{(i)}\}, \{\boldsymbol{v}^{(i)}\}\right) = \sum_{il} \left(z_l^{(i)} - \phi\left(\sum_k w_{lk} \phi\left(\sum_j w_{kj} v_j^{(i)} \right)\right)\right)^2 + \lambda_2 \left(\sum_{lk} \|w_{lk}\|^2 + \sum_{kj} \left\|w_{kj}\right\|^2 \right). \qquad \text{Equation 5}$$

The squared error between the labels and the feedforward outputs can be replaced by other more suitable measures (logistic regression etc.). Present-day deep learning architectures have 20+ layers instead of the two depicted above. Note that from a mathematical point of view, this presents a straightforward extension. However, the number of layers is often critical to performance and underlies the difference between under-fitting and over-fitting. The squared norm regularization terms are also standard (gated by a parameter λ_2) with other forms available as well.

The software suites available at the present time (TensorFlow etc.) largely automate the process of setting up the number of layers and choice of error measures. We applied these libraries in a standalone mode to deliver a deep network module and in feedback mode wherein the weights above are tuned within a larger objective function (containing the dictionary learning and Gabor wavelet terms).

We used a majorized version ([40], [41]) of the linear support vector machine in the output layer. We have considerable experience with training/testing majorized SVMs [Yan17, Yan17b] with and without wavelet preprocessors. In addition, we have enormous experience with over-complete representations and convex approximation algorithms for classification ([42], [43]) in general. The majorized SVM objective function is wherein auxiliary variables $\boldsymbol{s}$ have been used to transform

$$E_{\mathrm{SVM}}(\boldsymbol{w}, \boldsymbol{s}) \ = \frac{1}{2} \|\boldsymbol{w}\|^2{}_2 + C \sum_i \frac{\left[1 - y^{(i)}\left(\boldsymbol{w}^T \boldsymbol{x}^{(i)} + b\right) + s^{(i)}\right]^2}{4 s^{(i)}} \qquad \text{Equation 6}$$

the standard hinge loss objective function into a majorized version. The weight vector w and bias b are standard, as are the class labels $y^{(i)}$. (We have used a different notation for the labels here to differentiate this module from its deep learning counterpart.) The notation $x^{(i)}$ denotes the input patterns, which may be deep network outputs instead. This linear SVM can be readily extended to multiple classes using one-versus-all training. The parameter C, which expresses a trade-off between margin maximization and classification, will be set using cross-validation.

4.6 Deep Discriminative Classification

Machine learning methodologies for infrasound processing can be expected to result in classification algorithms ([44], [45]) that extract features from a variety of wireless infrasound signal sources. Assuming signal fusion is in place via adaptive beamforming and related approaches, this suite of methods proceeds with feature extraction (using convolutional networks and deep learning) [45] followed by efficient classification (using support vector machines, etc.) [44]. We have expertise in all of the above ([5]–[7]). However, machine learning is not sufficient for signal characterization: to discover transient signal signatures that are characteristic of the domain ($10^3 - 10^9$ kg TNT equivalent yields), we turn to deformable fragment templates.

When signals pertinent to a domain are collected, they are fused using dynamic time warping and related methods to remove non-stationary and local shifts [46]. Consequently, the fusion process has information regarding the temporal correspondences across the signal stack. Machine learning methods can (in their lower layers) yield features that are important to a particular domain [47]. The main difference between standard machine learning and infrasound processing is that we are dealing with low-frequency signals. We can convert the low-level features into temporal functions representing a particular domain (airborne blasts, for example). The lower layers (mainly convolutional) give us linear operators which transform the signals into feature signals. And since we have correspondence information from signal fusion, these new feature signals can be brought into the register using fusion information (following similar approaches in 2D and 3D) [48].

We have developed a processing, denoising, discrimination, and classification pipeline comprising Gabor wavelet signal representation, over-complete dictionary learning, deep networks, and culminating in a SVM classifier.

4.7 Data Collection

The ability to acquire and efficiently disseminate cyber-physical data is critical to the success of any research project targeting this domain. Due to high diversity and heterogeneity in the characteristics of smartphone sensors, exploitation techniques need to be more adaptive and flexible to transition seamlessly to the cyber-physical realm. With this consideration in mind, we worked directly with infrasound data collected via mobile device. Our team member RedVox, Inc., has unrivaled capabilities in this regard built on decades of experience interpreting infrasound signatures through the chain of signal capture, detection, association, localization, and identification. RedVox has an iOS and Android infrasound application that can leverage the ubiquity of mobile device microphones. RedVox has demonstrated ([49]–[51]) the suitability of on-board mobile device microphones and barometers to record down to the infrasound range of interest for tactical and explosion monitoring applications. RedVox also has successfully crowd-sourced data collections from various SpaceX launches out of Cape Canaveral, FL. RedVox also develops, tests, and implements cloud architectures for data centers.

The foundation for transient signal detection and characterization is a refined version of the [11] multiresolution gridding method to standardize spatiotemporal observations and feature sets into fractal logarithmic spaces for comparisons amongst different sensor modalities (INFERO). We recast the signature identification problem [52] in terms of the reproducibility and transportability of robust signature metrics and their variability. Even simple signatures, such as those produced by controlled explosions [21], can have substantial diversity in their defining features. The association and localization procedure depends on the selected target and the topology of the sensor network. RedVox has access to traditional sensor and array data methodologies incorporated with the emerging ubiquitous sensing systems to associate and locate sources based on shared signature features and probabilistic variability. We concentrate on migrating, updating, and implementing traditional and emerging sensor systems and methods into Big Data architectures.

4.7.1 Data and Metadata Quality Gates

The distinction between data and metadata can be blurred once the processing chain is initiated and additional information layers are added (metadata). In this discussion, data refers to the binary data payload, usually corresponding to time-series data points, and metadata corresponds to information stored either on RedVox headers, CSS tables, or dataless volumes. It is assumed that concurrent data packets are streaming continuously from different heterogeneous sensor systems. For the purposes of this discussion, the existence of the data becomes relevant for a specified time duration and region of interest. In other words, we are most interested in ensembles of data packets at a prescribed nexus of space and time. The highest value is placed on data packets corresponding to the expected arrival time of a signal at a given range from the source with a specified minimum signal to noise (SNR) ratio within a frequency band of interest. The first constraint (expected travel time) is an assertion of causality. The second constraint (SNR) is a requirement for a modicum of usable information above the noise. Adaptively defining noise relative to a specific signal of interest was a key RedVox task.

The metadata provides some measure of the stability, stationarity, and timeliness of the data. Of particular interest is the location of the measurements and their variability. This parameter is very important in the case of a moving observer. Furthermore, the causality constraint cannot be enforced without location information. Although in principle, it is possible to scan all data for specific feature sets, the return on computational investment may not be worthwhile. The metadata also has key information on timeliness and the precision that can be associated with arrival times. Thus, the position and timing metadata information should be handled with exceptional care in the case of spatiotemporal data sets.

Data had various defects during a prescribed time interval. These include:

- Non-varying values (constants), such as zeros.
- Data gaps, either from sensor malfunction or dropped packets.
- Not-a-number (nan) or infinity (inf) data points.
- Clipped values (maximum dynamic range) may include legitimate transients.
- Spikes (unclipped) may include legitimate transients
- Harmonics, instrumental, or environmental. It may contain calibration tones.

For some types of data, such as temperature or barometric pressure, constant values may be acceptable. However, in these instances, it is worth reducing the dimensionality of the data payload so that it can be represented simply and concisely. The effects of each of these possible defects in the data and metadata need to be measured and tracked, as well as the cumulative effect of these defects in a measure of conformance to specifications.

If the data is unusable, a trace of its presence should be stored with a record of the rejection criteria. These unusable data should not move forwards to the processing chain. Acceptable data should also have relative figures of merits, as only the best should be selected for ML training sets. Questionable but usable data can potentially yield arrival times, signal bandwidth, transmission loss, and other usable parameters. Of importance is to quantify what proportion of all the available data within a prescribed time interval and region of interest is usable. The expected network attrition rate is important to be able to estimate or forecast how many devices would be needed to provide a high probability of detection. The stage for initiating the processing chain with the subset of usable data, with causality and SNRs as primary signal arrival discriminants and data quality metrics for usable channels.

4.7.2 Selecting the Best Data

The problem of selecting 'good quality' data within the subset of 'usable' and 'acceptable' data has similarities with detecting domains with trustworthy content where one must discriminate between credible and spam sources.

Conformance will depend on the specifications of the intended processing chain. For example, timeliness and position stability would be a requirement for source localization. The sensor location should be the 'best’ estimate over the period of inspection. For a selected analysis time segment, the mean, median, max, min, and standard deviation of the location should be returned, as well as some measure of the timeliness of a data packet. For the RedVox packets, this information is contained in the statistics of the metadata header information. Basic network source localization to provide seed locations can be performed without ML. However, high-confidence source localization should incorporate some level of signal classification to reduce false associations. Once high-conformance data are selected for signals of interest, we can move towards feature engineering for ML.

5 Results and Discussion

5.1 ML Model Development, Deployment, and Validation

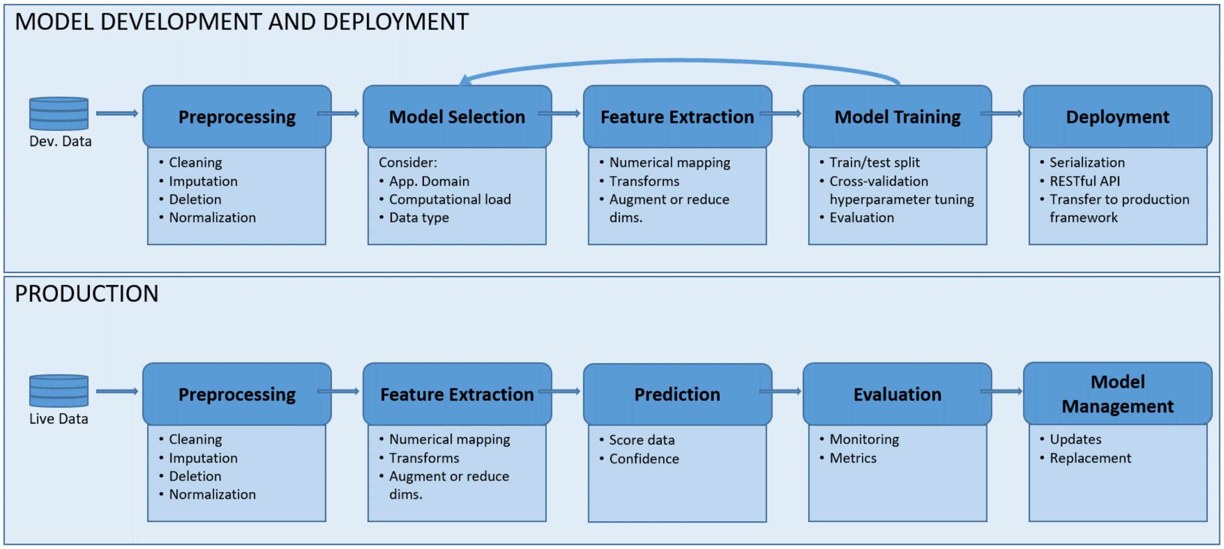


Figure 2 Machine learning process.

Our team has proven and demonstrable expertise in implementing, deploying, and effectively assessing the performance of machine learning (ML) systems. Transitioning research, proof-of-concept pilots to the production scale analysis required by the MASINT-X program requires a vetted process and fluency with modern big data processing frameworks. The ML *implementation and deployment* process is designed to support the exploratory and iterative characteristics needed to develop robust models. It dovetails with our *production process* that ensures on-the-wire prediction with consistent performance and sustainable management. The two processes are illustrated in Figure 2. Our ML models are subjected to rigorous performance assessments through a variety of evaluation metrics (e.g., precision, recall, ROC curves, etc.)

([53], [54]), ensuring repeatable scoring accuracies critical to operational mission systems. In what follows, we detail aspects of our ML system development, production, and performance evaluation processes.

5.2 ML Implementation and Deployment Process

The implementation and deployment process begins by creating the necessary *preprocessing* operators needed to make the development data ready for data mining. This includes actions such as cleaning, imputation (filling missing data), deletion, and normalization (e.g., controlling dynamic ranges of attributes). In addition, resampling strategies (ensuring proper class stratification) are used to alleviate class imbalance issues [55]. After scrubbing the data, we commonly enter a three-step iterative process to determine the most suitable ML model for the application domain of interest. The steps involve model selection[1], feature extraction, and model training. During *model selection*, system stakeholders and ML experts took a variety of decision parameters into consideration to select the most suitable ML algorithm. Decision considerations include the unique characteristic of development and production domains, data type(s) of interest, computational loads, the use of ensemble versus individual models, fusion strategies, etc. Next, the *feature extraction* step transforms the preprocessed data to a format best suited for the data and chosen model. This may involve conversion to numerical representation (e.g., text to term-frequency), transformations (e.g., Fourier transform for signal data), augmenting or reducing features (e.g., principal component analysis), or perhaps not modifying the data (identity transformation). With the proper feature representation in place, the next step fits a model—commonly referred to as *model training*.

Model training has several critical processes that strengthen our ability to produce quality predictions: train/test splitting, cross-validation for hyperparameter tuning, and evaluation of the model's generalization capability [56]. The train/test split partitions our development data into a training and test set. Typically, 60/40, 70/30, or 80/20 training to test ratios are used. We ensure that the split retains any stratification properties to ensure proper class balance. The training data is typically further partitioned into a k-fold cross-validation set to support hyperparameter tuning. Hyperparameters are any free parameters selected prior to fitting the model, e.g., the number of trees in a random forest classifier. The cross-validation set is used to select the optimal set of hyperparameters. During the fitting of the model, i.e., the parameter estimation, it is imperative the proper mathematical optimization technique is selected to achieve the most optimal set of learning weights (e.g., use of convex optimization, proper regularization, etc.). Once the tuning and fitting are complete, the model is evaluated on the unseen test data to gauge a final overall measure of performance. We use a variety of *modern performance metrics (precision/recall, sensitivity/specificity, ROC curves, etc.)* that go beyond simple accuracy to obtain a model with robust prediction characteristics (see Section 2.6 for more details).

Once the model is fit, it is ready for deployment to the production system. There are a variety of deployment frameworks, and making the choice of the best-suited one must take into account factors such as the model itself, software frameworks employed (backend, caching, web service), latency, etc. Our team's collective expertise extends across many of the most popular solutions like predictive modeling mark-up language (PMML [57]) and serialized access through custom web services (e.g., pickling in conjunction with RESTful APIs [58]). When considering an

[1] Our use of the term 'model selection' in this context to simply refer to the choice of ML algorithm should not be confused with how it can also be used to refer to the hyperparameter tuning process involved in the model training step.

operational environment with high-throughput demands (like the present mission), our deployment solutions consider the use of dedicated clusters for ML model processing, supporting frameworks to handled load balancing and caching, and proper updating of existing models, taking into account pitfalls like library inconsistencies.

5.3 ML Production Process

In the production environment, the unseen live data passes through the same *preprocessing* and *feature extraction* steps settled on during the machine learning model development phase. It is then ready to be scored by the deployed *prediction* engine (model). Depending on the model, the predictions may provide associated confidences—a desirable output for compliance justification. The prediction model is continuously monitored through the calculation of performance metrics by the *evaluation* step. These performance characteristics are inputs to *model management*. The management suite handles decisions that may result in the model requiring an update or total model replacement. A model update requires returning to the development testbed. However, the update process circumvents the need to perform the iterative cycle used during development. We only employ the model-training step to re-train a new version of the current model, augmented with additional data obtained since the last update. Then, the deployment step is used to send the model to production. If a total replacement of the model is necessary, we may need to return the full development cycle, augmented with additional data. However, this process is usually more streamlined, given the experience gained during operations. The production process ensures that the overall system minimizes downtime, ensuring fault tolerance, and uninterrupted update-replacement procedures.

5.4 ML Model Validation

The ability of ML models to achieve the desired levels of operational performance is a function of the *performance measures (evaluation metrics)* used to assess their effectiveness during training and trial production runs. Often, casual users of pre-packaged ML libraries make the false assumption that the use of a particular evaluation metric is equivalent to any other. With years of building and deploying ML models, our team is aware of the pitfalls associated with the various classification performance measures. Here we detail several of these metrics and provide insight as to how well they predict the performance of ML models. We will discuss the performance measures within the context of a simple two-category classification problem, with the understanding that all discussions readily extend to multi-class scenarios. These measures are well known in the ML community; however, their misuse and underutilization when developing practical, mission-critical systems have often led to suboptimal ML models and, consequently, overall poor system performance.

The *confusion matrix (contingency table)* is an organized way of capturing the labeling prowess of the ML model. It reflects correct and incorrect classification across both classes of interest, as shown below.

Table 2 Confusion matrix definition.

	Prediction Label: Class I	**Prediction Label: Class II**
True Label: Class I (Positive)	True Positives (TP)	False Negative (FN)
True Label: Class II (Negative)	False Positive (FP)	True Negative (TN)

From the confusion matrix, we can derive the following popular evaluation metrics (not exhaustive):

- True Positive Rate (Sensitivity) $= \frac{TP}{TP+FN}$
- True Negative Rate (Specificity) $= \frac{TN}{TN+FP}$
- Accuracy (1-misclassifcation error) $= \frac{TP+TN}{TP+TN+FP+FN}$
- Precision $= \frac{TP}{TP+FP}$, Recall $= \frac{TP}{TP+FN}$

Many inexperienced practitioners simply rely on accuracy as a measure of performance. However, using accuracy alone has some serious disadvantages. It does not perform well in situations with significant class imbalance issues. It is easy to maximize accuracy under this scenario artificially, have a decision rule that labels all data as belonging to the class with the larger population. Such a system is of no practical use, but this illustrates how accuracy alone is not the best indicator.

Precision and recall focus on the number of true positives to address the more pertinent question: What percentage of the relevant classes have been labeled correctly and as well as the number of false positives? In certain applications, one may be solely interested in high precision, i.e., only being concerned with observing a small number of events and desiring that as many of these as possible are labeled correctly. For the present cyber-physical classification application, intelligence analysts will more than likely be concerned with having a high recall value, even if this results in low precisions. With a high recall for event matching, we would label as many of the relevant events as possible from the captured data. However, we must be careful: a recall of 1 can be achieved by simply labeling all signal captures as the event of interest. This would ensure we get back all the event signals. But, as precision and recall trade-off against one another, the precision value for this scenario would be very low. The recall is non-decreasing in terms of the number of events classified. In a balanced ML system, precision will decrease as we increase the number of events we attempt to label. As a rule of thumb, we typically want a good recall within some tolerance of false positives.

Precision and recall can also be compared to receiver operating characteristic (ROC) curves. Before discussing ROC curves, we first define two other statistical measures often used in classification performance evaluation: sensitivity and specificity. Sensitivity (aka true positive rate) is identifiable with the previously defined recall value and measures the total number of positive events (i.e., events of interest) that were classified correctly. Specificity (aka true negative rate) measures a total number of negative (or non-relevant) events that were correctly classified as negatives. A ROC curve is a graph of the sensitivity (true positive rate) versus 1-specificity (false positive rate). Each time we run a classifier (e.g., under various threshold values), we get a confusion matrix. Each point on the ROC curve represents a realization of a confusion matrix. If we have a case of a perfect classifier, then both sensitivity and specificity will be 1. Hence, the best predictor would evaluate to value in the upper left corner of a ROC graph. A classifier is unable to discriminate, equivalent to random guessing, would result in a point that lies on the diagonal in the ROC space.

The metrics we have discussed to this point are based on ML classification models that assign a single label from a set of discrete possibilities. There are many ML models that provide a ranking rather than a single label. This enables us to delve deeper into the performance of the model through the use of rank-based metrics [54]. Rank-based evaluation metrics, such as Mean Average Precision (MAP) ([59], [60]), focus on analyzing the order of labeled confidences (or distances). Intelligence analysts often find rank-based results more meaningful since they allow experts to evaluate multiple, closely related classification results rather than relying on a single label.

Our team's familiarity with these and other evaluation metrics ensures a rigorous assessment of all ML models developed and deployed during this project.

5.5 Open-source Development Platforms for ML Data Processing and Analysis

At a high level, the system acquires infrasonic data from mobile devices and then attempts to characterize the signal source and apply ML models to classify events of interest. The data acquisition is handled by our team member RedVox, who has developed a cloud-based, fully scalable framework to handle simultaneous recording by numerous users. When the system receives data faster than it can process (commonly known as backpressure), the AWS framework throttles the incoming streams using Apache Akka and Kafka. This improves the data flow by regulating the ingest rate, allowing for reduced system response times and the processing of large data sets. Akka alleviates the backpressure, while Kafka allows buffering of data bursts.

The ML models are implemented using a variety of open-source analysis engines and development environments. Custom algorithms are developed in Python, leveraging its ecosystem of data analysis libraries like pandas, scikit-learn, SciPy, NumPy, etc. The TensorFlow framework is ideally suited for neural architectures, supporting distributed training on clusters and a variety of optimization algorithms. The DFL and CE engine is well suited for this platform. The already existing tight integration between Python and TensorFlow minimized development delays and ensured that the D3IC capability is scalable and robust from prototype to production.

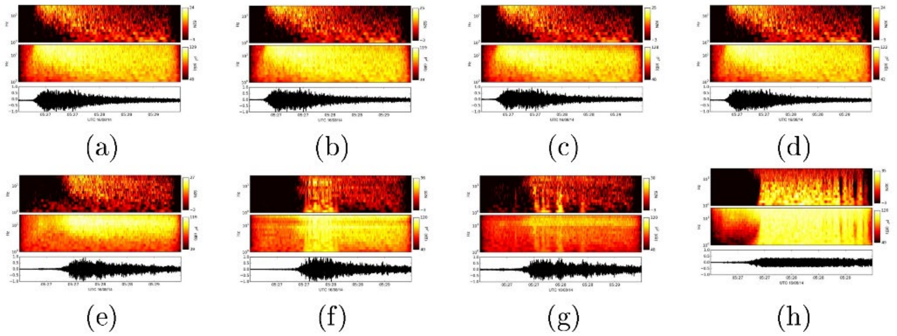


Figure 3 Comparison of high-fidelity infrasound data collected via sensor array versus mobile device.

To demonstrate the need for state-of-the-art machine learning and signal processing algorithms customized to the cyber-physical domain, consider the data collections illustrated in Figure 3. These are collected infrasound recordings from the Falcon 9 rocket launch from Cape Canaveral, FL. The observations were simultaneously collected by a high-quality array of Chaparral infrasound sensors and by a set of iOS devices (iPads and iPhones). The first row in Figure 3 (a)-(d) contains the data obtained by the sensor array, located within 5 km of the launch site. Each sub-figure has three panels: the top panel corresponds to SNR, the middle panel represents spectral power squared in decibels, and the bottom panel contains the raw, recorded waveform data. The second row in Figure 3 (e)-(h) contains the data from iPhones and iPod Touch units located within a 14 km radius of the launch site. All data were recorded within a frequency band of 0.5-32 Hz.

The traditional sensor data near the source exhibits more uniformity in the waveform amplitudes and spectral characteristics. This readily allows the data to be processed by traditional algorithmic techniques to improve SNR, then perform detection, characterize physical phenomenology, and provide event classification via standard machine learning approaches. On the other hand, the data from the mobile sensors further afield exhibit many irregularities due to effects such as diffraction, dispersion scattering, and non-linear shocks that are compounded in the cyber-physical domain (as expected since the multipurpose device is not optimized to capture this type of data). This test dataset clearly illustrates that the same event observed in the traditional domain near the source can appear considerably different from a cyber-physical device at a distance. Recent data collections using selected Android smartphones have demonstrated improved system response approaching traditional systems.

5.5.1 Data

5.5.1.1 Firearm Data

To augment our dataset, we constructed a synthetic library of firearm signals. This allowed our team to successfully process firearm samples (around 735) and perform preliminary dictionary learning and principal component analysis (PCA). Several examples of typical firearm waveforms are shown below in Figure 4.

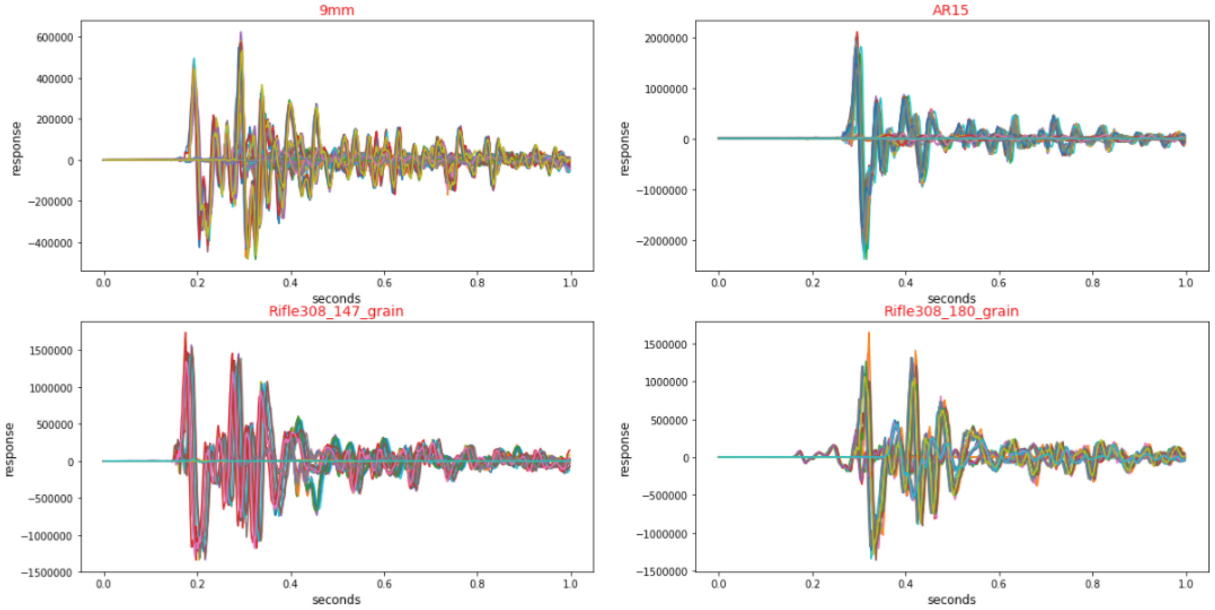


Figure 4 Sample firearm acoustic signals.

The Figure 4 distribution of samples per gun are (i) 9mm: 236, (ii) AR15: 300, (iii) Rifle308_147 grain: 100 and (iv) Rifle308_180 grain: 100. The shots were detected and boxed into 1 second windows. These samples were then mapped into the frequency domain (using FFTs). All subsequent analysis was performed in the frequency domain. Typical examples of frequency domain information are shown below. The Fourier magnitude information is shown (which is symmetric in the frequency space).

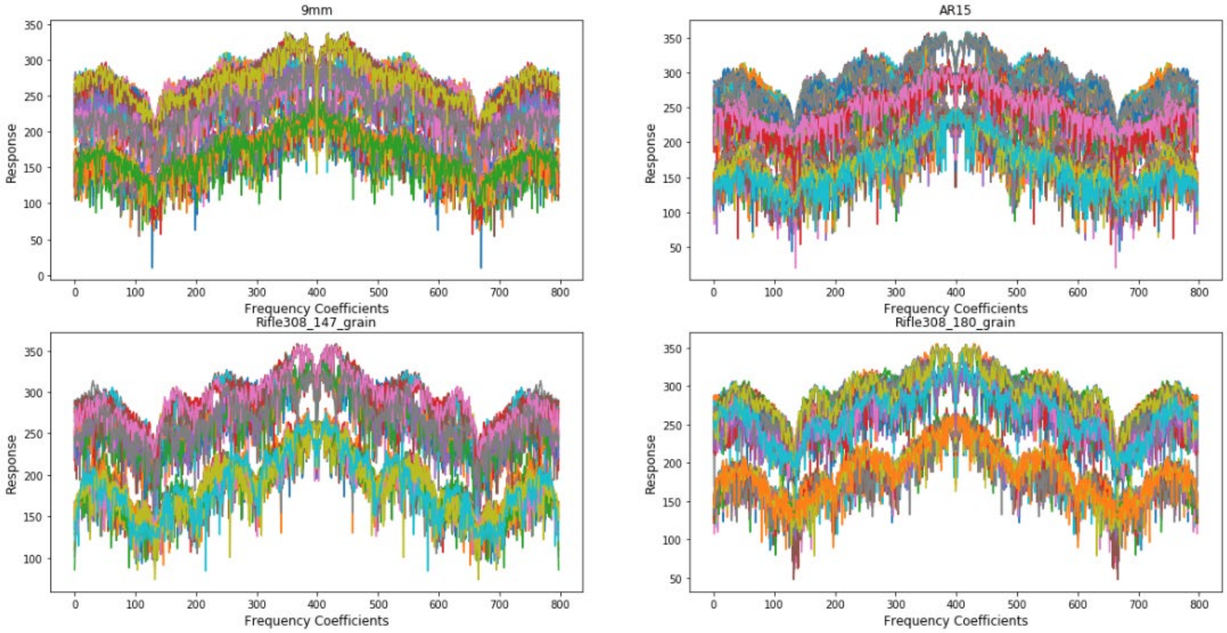


Figure 5 Frequency domain visual of the firearm signals.

5.5.1.2 UAS Dataset

An initial analysis of a UAS dataset was considered. We took some summary statistics of the data and identified general patterns. We then inferred some information about the data given some description of the collection process, location of the devices, and the flight pattern. When devices were inside a vehicle, there was an audible difference in the signature, that was easily recognizable in the Fourier domain. There was optimism that this difference will hopefully be captured in the generalization of our model moving forward. Figure 6 shows the Fourier Transform plot of two cases of a drone.

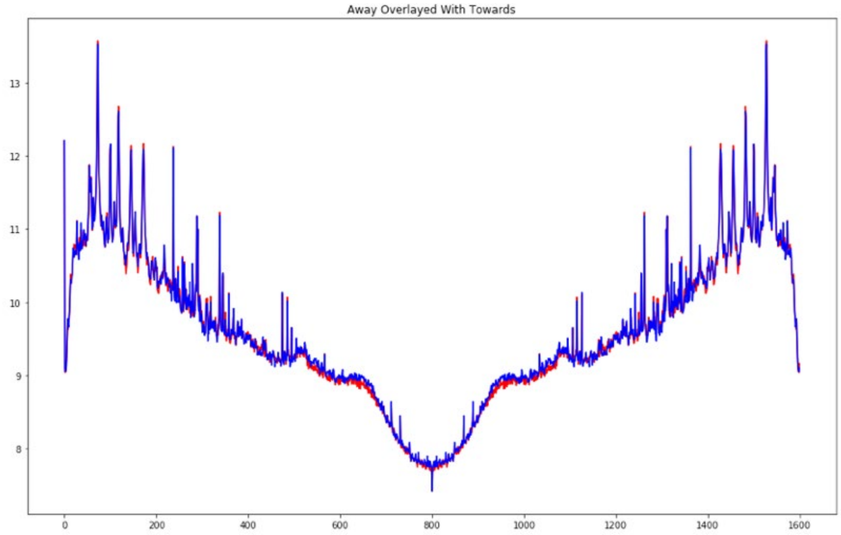


Figure 6 Log Fourier Transform of sounds signature of drone flying away (red) overlaid with drone flying towards (blue).

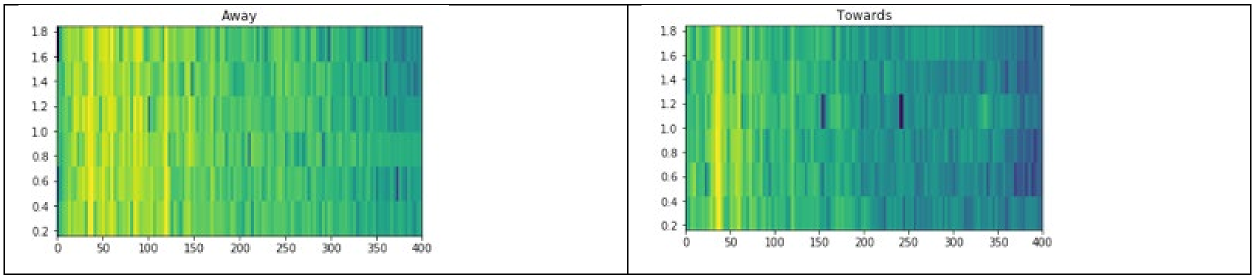


Figure 7 Spectrograms with .32s windows of away and towards drone signatures.

5.5.2 Robust Feature Extraction

We first constructed a filter bank of Gabor filters by creating an $m \times\ n$ matrix **A** with n equal to the number of atoms and m equal to the length of the signals we are encoding (in this case 799). For the number of atoms, we chose to have one Gabor Filters centered around each possible shift, with sigma's varying according to $f(x) = \exp(x)$ where $x = -5, -4, \dots, 4, 5$. So for each possible shift, there are ten different width Gaussians that are centered there.

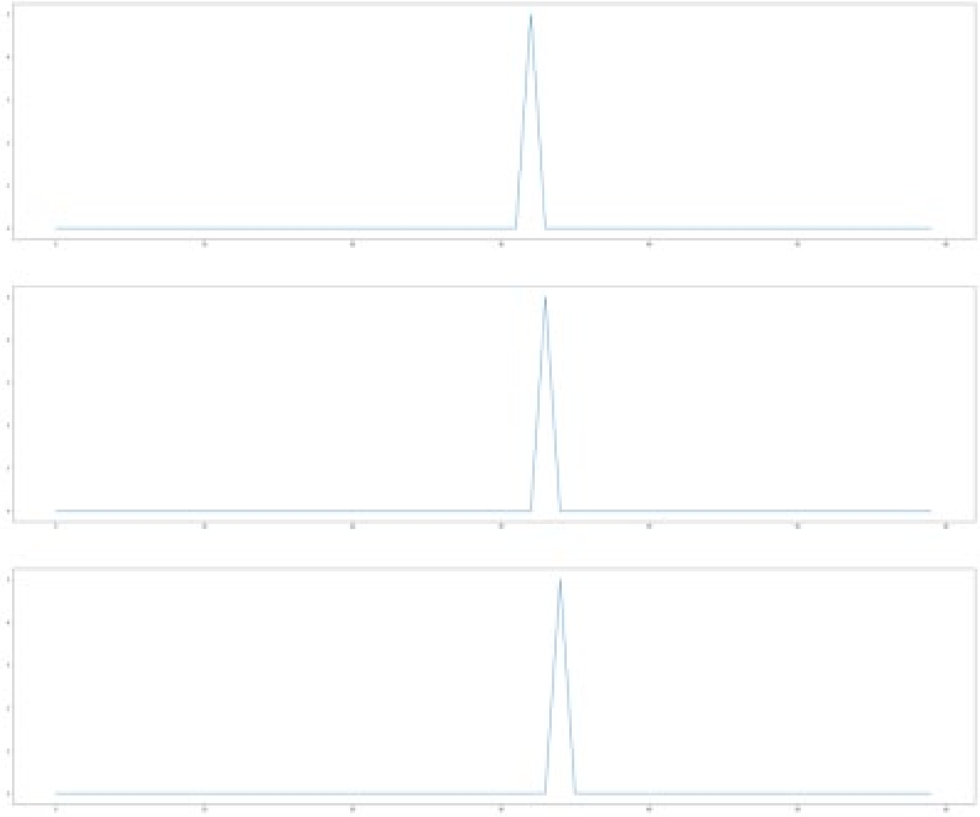


Figure 8 Three Gabor filters with the same width at 3 different shifts.

We then take this Gabor Matrix and use a sparse coder algorithm to code each signal into a constrained number of combinations of a nonzero coefficient. The final code is a $n \times 1$with vector with only k nonzero elements. Figure 9 is an example of a coded signal.

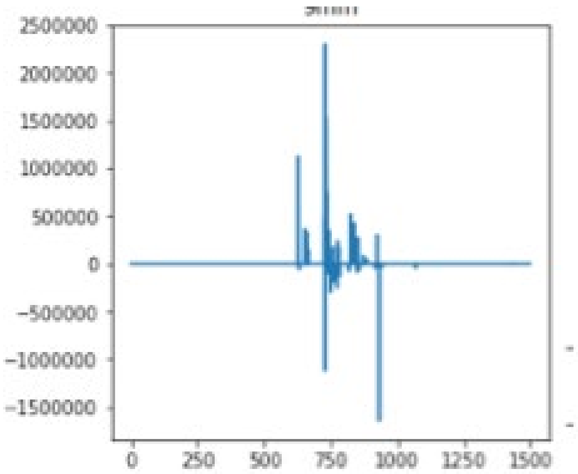


Figure 9 Example of a sparsely coded 9mm signal.

We then use these coefficients as features in our classification algorithm and get around 80% accuracy using a 30/70 split and 90% accuracy with a 70/30 split, running through PCA and an SVM classifier.

5.5.3 Classification Techniques

We performed principal component analysis (PCA) on all firearm samples. Examples of PCA "atoms" are shown below in Figure 10. Again, all analysis is performed in the frequency domain, as mentioned above.

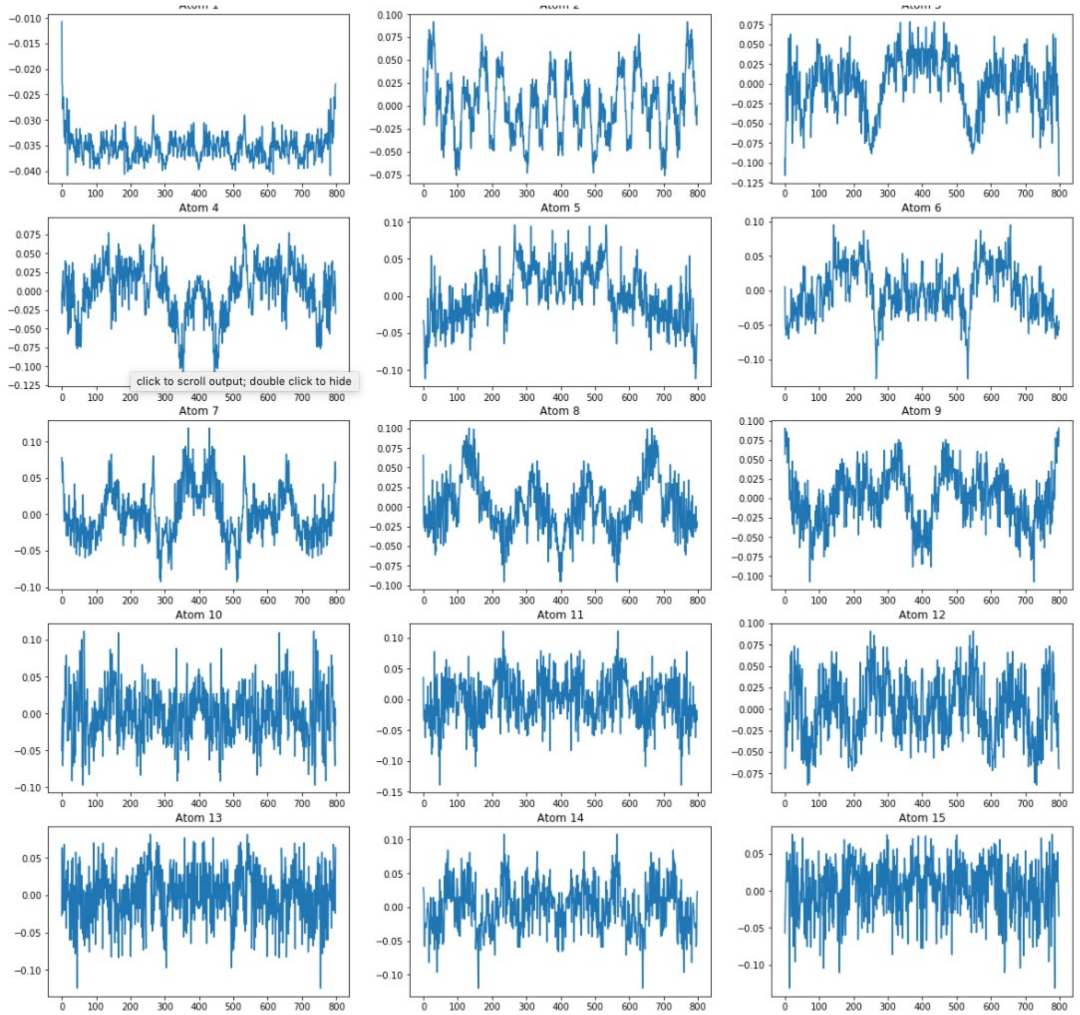
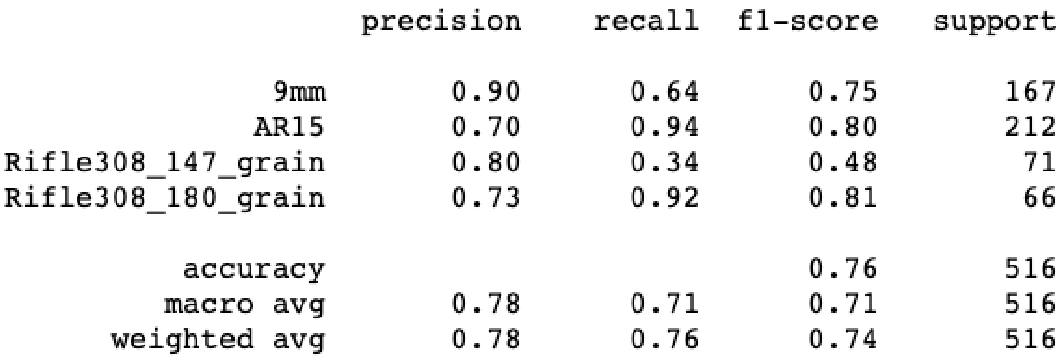


Figure 10 PCA atoms of signals.

Next, we executed the K-SVD dictionary learning algorithm on all gun samples. Since the “atoms” are overcomplete, it is not easy to estimate each atom's relative importance. Some examples (again in the frequency domain) are shown below. These results are anecdotal and more work needs to be done to establish the efficacy of dictionary learning.

5.5.3.1 Kernel Discriminant Analysis

This KDA algorithm was developed by [61]. The algorithm is known for producing state-of-the-art results compared to other dimensionality reduction methods like PCA, PCA-LDA, and kernelized methods like SVM. The results for this algorithm in conjunction with a KNN on the firearms dataset are shown in Figure 11.



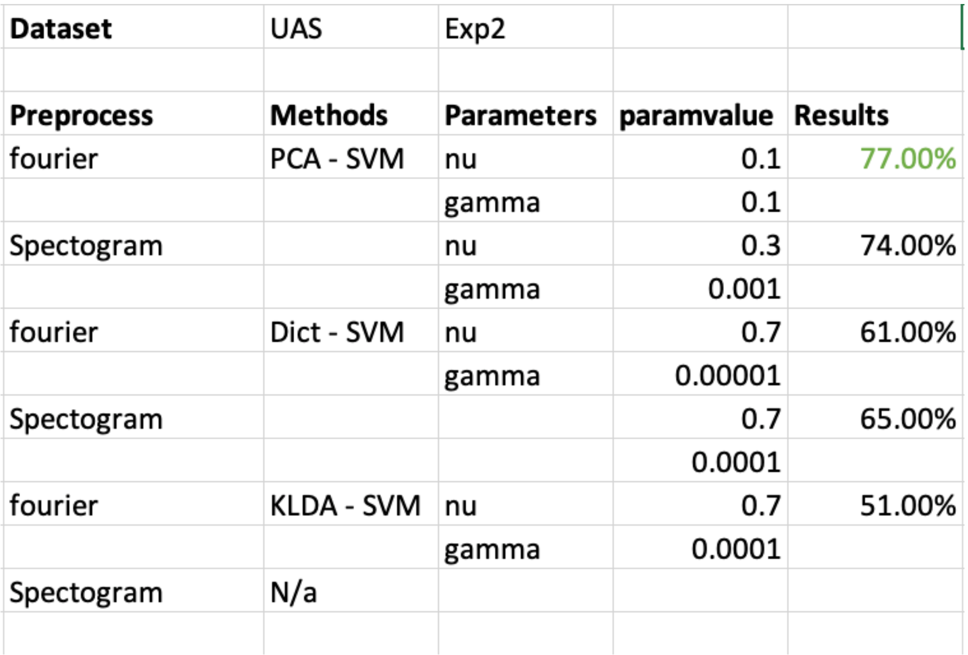
```
                      precision    recall  f1-score   support

                9mm        0.90      0.64      0.75       167
               AR15        0.70      0.94      0.80       212
Rifle308_147_grain        0.80      0.34      0.48        71
Rifle308_180_grain        0.73      0.92      0.81        66

           accuracy                            0.76       516
          macro avg        0.78      0.71      0.71       516
       weighted avg        0.78      0.76      0.74       516
```

Figure 11 KDA classification report between the four firearms classes using a 70% training 30% test split.

The classification results for KDA have a higher accuracy than dictionary learning classification. The dictionary learning experiments produced an average of only 67% classification accuracy, while PCA produced a closer 94%. This was unexpected because as we know supervised methods look to maximize the class separation and minimize the in-class variance. PCA picks combinations of features that explain the most variance. One explanation of this is the dataset's size, where it has been shown that PCA can outperform LDA when the dataset is small.



Dataset	UAS	Exp2		
Preprocess	**Methods**	**Parameters**	**paramvalue**	**Results**
fourier	PCA - SVM	nu	0.1	77.00%
		gamma	0.1	
Spectogram		nu	0.3	74.00%
		gamma	0.001	
fourier	Dict - SVM	nu	0.7	61.00%
		gamma	0.00001	
Spectogram			0.7	65.00%
			0.0001	
fourier	KLDA - SVM	nu	0.7	51.00%
		gamma	0.0001	
Spectogram	N/a			

Figure 12 Our ML pipeline results on UAS data.

5.5.3.2 PCA --SVM

Our team focused on running the full UAS dataset through the machine learning pipeline and getting baseline results for drone direction classification. Our results for determining the drone's direction (away or towards) at any given time interval based on audio signals were not clear. Our initial baseline using our pipeline, which includes taking the PCA scores and sparse coding signatures from dictionary learning into a non-linear classifier, produced reasonable results.

Several experiments focused on expanding our classification pipeline to optimize our current preprocessing and classification parameters on all the different RC helicopter datasets. We sought to understand if our methods achieved significantly different results per experiment to reveal any discrepancies in each experiment's difficulty. We also added a new preprocessing step of taking a spectrogram of the signals and passing the coefficients of those as features into our classifiers. The spectrogram was first *flattened*, meaning if the spectrogram had *m* size FFTs and *n* time-intervals, the flattened spectrogram would just be one row of *nm* entries. In this format, the entries would be fed into some classifier to determine if an RC copter was flying towards or away. We continued to use the successful PCA into SVM and Dictionary Learning into SVM classifiers that we had built earlier. We then used different window sizes of our signals to see if taking specific length samples helped classify. Lastly, we generalized each model to the full dataset to understand how well the models learned RC copters' discriminative features flying towards and away.

Our model results scored highest on the validation sets, with the tuned PCA – SVM model producing the highest score of 77% classification accuracy. Flattened spectrograms with 1/3s windows also produced comparable results. Although 77% is good, the classification accuracy seems to depend on the length of the interval used to determine the direction that the drone is flying. With an interval size of .5 or .25 seconds, the classification accuracy nears guessing (50% for a 2-class classification problem). KLDA performs poorly on this dataset, as the method for computing kernels can be intractable for large datasets. We have omitted the experiments and will continue to investigate a more tractable way for numerical computation.

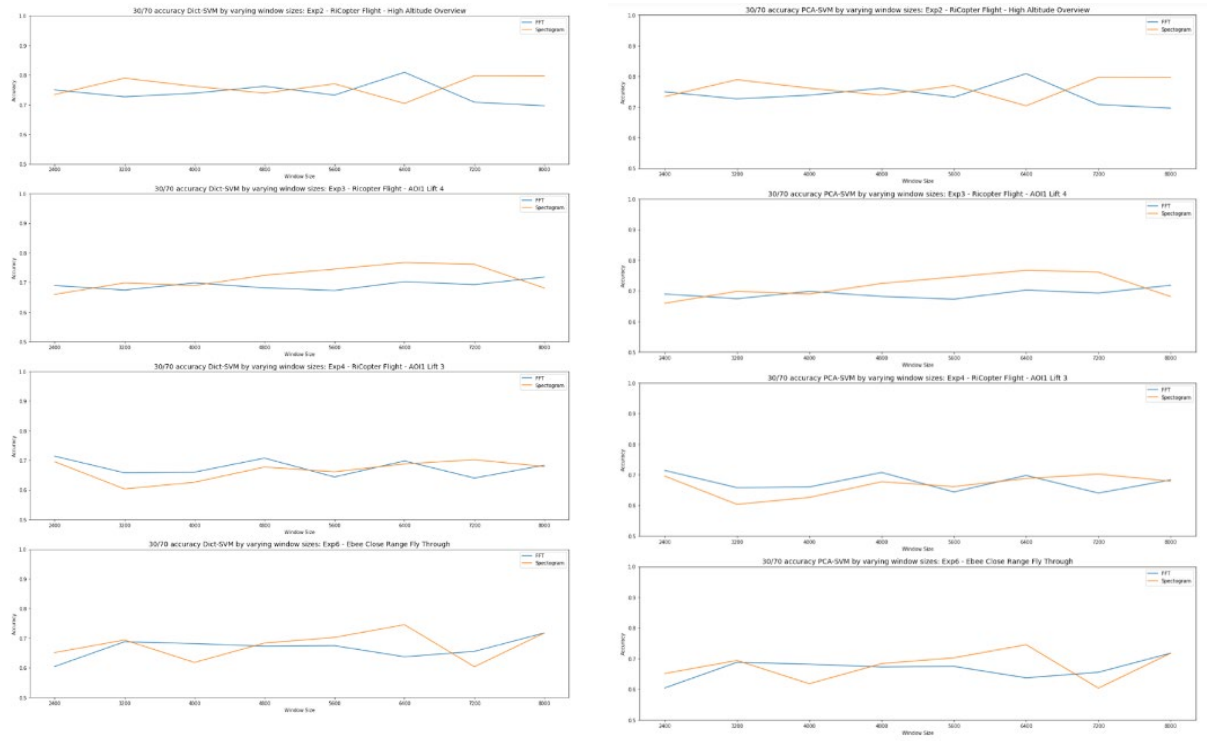
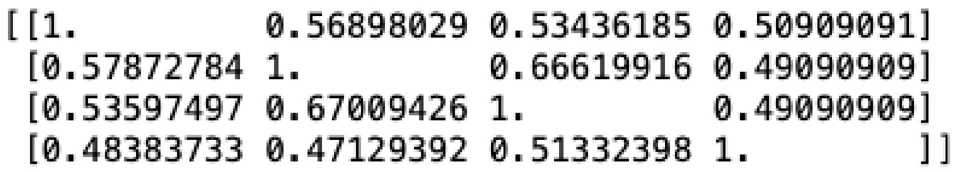


Figure 13 The PCA-SVM technique with the FFT feature accuracy (Blue) and the Spectrogram feature accuracy (Orange).

The results of this experiment are shown in Figure 13. On the left, results are shown for dictionary learning using a 30/70 split. Accuracy hovers in the range of 70-76%. For PCA in the same split, accuracy hovers around 81-86%. The best window ranges tend to be higher, around 8-10 seconds, but these window ranges are not always optimal as lower window sizes performed well in various experiments. The variance between the experimental accuracies between experiments was only 3% at worse. FFT outperformed spectrograms by more than one standard deviation, with an average performance bump of 5%. We presume this is overfitting because the algorithm memorizes the change in Doppler frequency of a towards/away copter and uses that difference for classification. This can be confirmed by the 70/30 splits showing significantly higher accuracy scores; scores for the 70/30 splits can be seen below.

On the right, we can again see PCA ahead of the dictionary learning method. When we generalize these models to the other experiments, we can see that performance is not that great. Below, in Figure 14, is the confusion matrix for the PCA-SVM experiment. The models perform significantly worse (almost 50%, which is near guessing). However, when the models are run on their training data, they achieve 100%, which is a sign of overfitting.



```
[[1.         0.56898029 0.53436185 0.50909091]
 [0.57872784 1.         0.66619916 0.49090909]
 [0.53597497 0.67009426 1.         0.49090909]
 [0.48383733 0.47129392 0.51332398 1.        ]]
```

Figure 14 The confusion matrix for the PCA-SVM experiment.

5.5.3.3 LDA-SVM

We used machine learning techniques LDA-SVM and a CNN on only the infrasound data from the gunshots. Our approach in this iteration utilized signal processing techniques that emphasize more discriminative features seen in Figure 15.

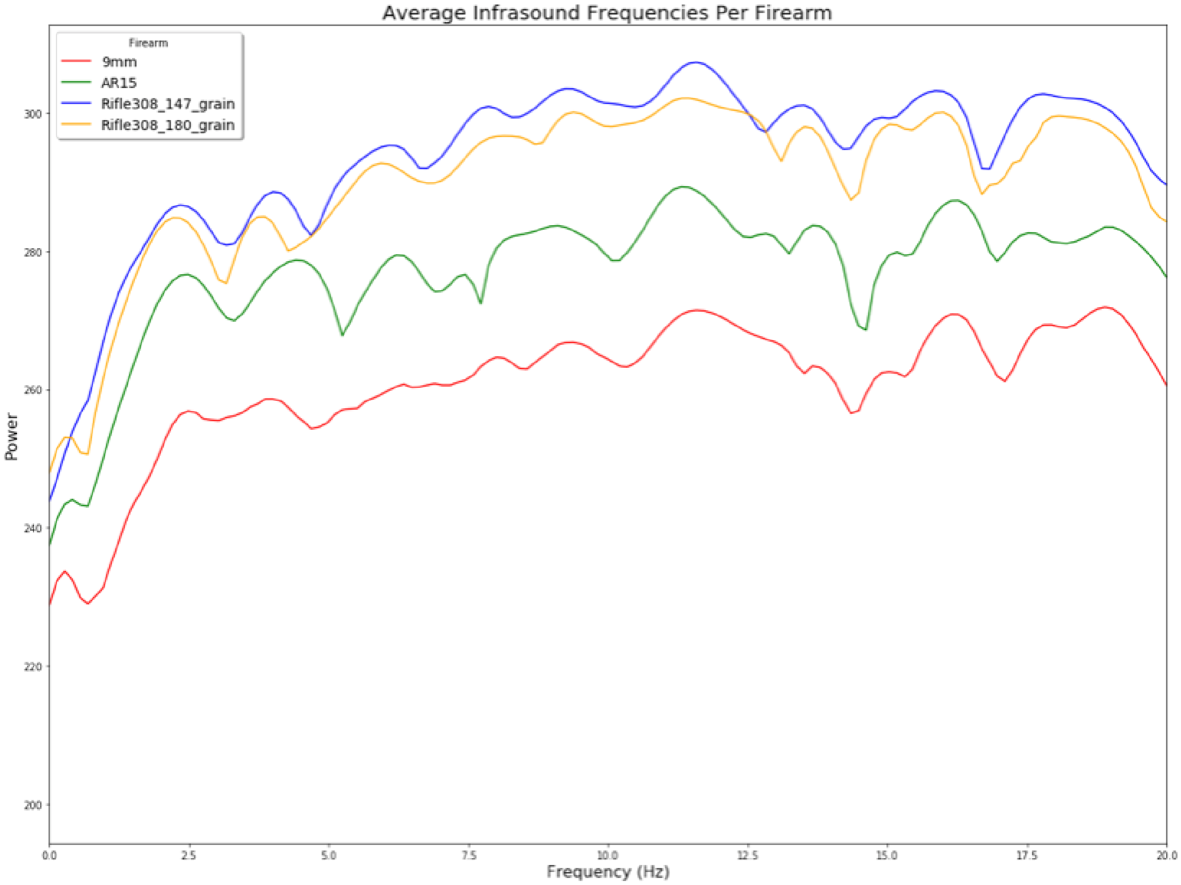
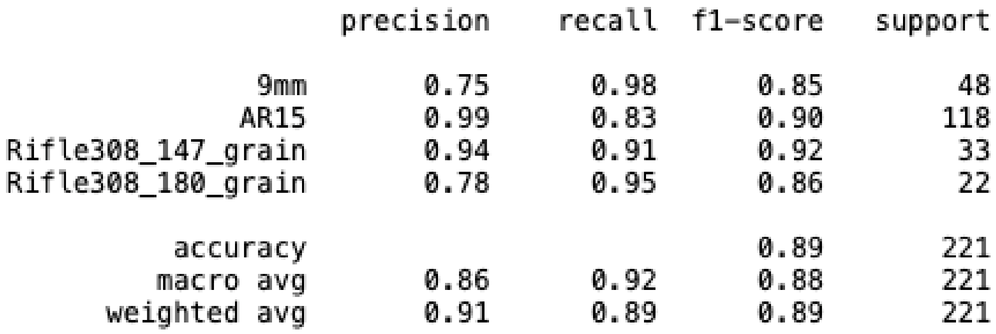


Figure 15: Infrasound frequency range of different gunshots.

In Figure 15, we took the signal average of each of the different types of guns and computed the log-scaled DFFT. We then used the frequency coefficients corresponding to the infrasound range (0-20Hz). The average energy at almost all of the infrasound frequencies is unique, indicating a measurable difference in the infrasound signature of each gun. The Rifle_147 grain and 180 grain are closer in resemblance, most likely due to the same gun. We performed a baseline classification using a Linear-Fischer Discriminant to project the data into a lower space

(c-1 dimensions, c being the number of classes). The data was then classified with a SVM using an RBF kernel. The results for a 30/70 validation split are shown in Figure 16.



```
                   precision    recall  f1-score   support

               9mm       0.75      0.98      0.85        48
              AR15       0.99      0.83      0.90       118
Rifle308_147_grain       0.94      0.91      0.92        33
Rifle308_180_grain       0.78      0.95      0.86        22

          accuracy                           0.89       221
         macro avg       0.86      0.92      0.88       221
      weighted avg       0.91      0.89      0.89       221
```

Figure 16 The classification results for LDA-SVM.

5.5.3.4 Leveraging Deep Learning

Our team leveraged some of the latest deep learning techniques for audio classification to compare to more traditional machine-learning methods. However, we found that the sample size that we attempt to train the network is too small, as our network attempts to learn thousands of parameters. We used a convolutional neural network with the same overall structure as in Khamparia et al. [62] on spectrograms of *only* the infrasound frequencies. This proved to be challenging, as there were not enough windows to make an image with a window size large enough to capture infrasound waves. We remedied this by padding each window before computing the FFT. This allowed us to have many windows with window-sizes that captured a modest amount of infrasound coefficients. The resulting spectrograms appear in Figure 17.

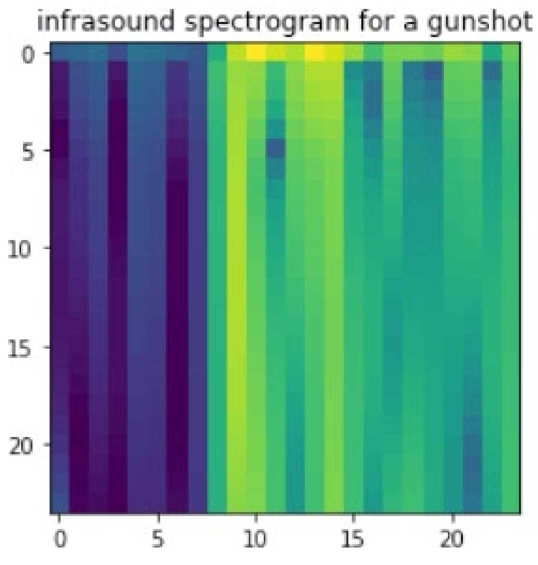


Figure 17: Spectrogram; x axis is time, y axis is power (log scaled)

One interesting thing to note about this spectrogram is the noticeable increase in power when the gun is fired in between windows 5 and 10. The exact pattern of the infrasound from the gunshot is what we had hoped our convolutional network learns as a basis for discrimination.

However, our network estimated too many parameters for a sample size of only around 500, whereas in Khamparia et al. [62], the sample size was around 50,000. Our results are better than guess, with an accuracy score of around .33 after 10 epochs. There is a possibility that our model is too large and is therefore overfit.

6 Conclusions and Recommendations

To summarize, we list high-level conclusions and recommendations gathered throughout the project:

Conclusions

- Convolutional deep learning models are robust to many signal preprocessing, and hyperparameters choices; recurrent deep learning models are less robust across all measured spaces.
- By utilizing acoustic/infrasound deep learning models enjoy improved classification performance.
- D3IC models outperform all other models due to dictionary learning; when discriminatory wavelet features are extracted, classification is based on these features to produce optimal results.

Recommendations

- Continued investigation of modern convolutional model architectures is highly recommended.
- Continued investigation of dictionary learning to produce a basis representation can represent waveforms as a set of feature coefficients.
- Test cross-domain transferability of features from gunshot and UAS datasets sources.
- Commission synthetics production pipeline to utilize more sophisticated physics and acoustic/infrasonic environment models.

7 References

[1] K. Smith, K. Bryan, M. Solomon, A. O. Smith, and A. O. Peter, "Near-field Infrasound Classification of Rocket Launch Signatures," in *SPIE: Chemical, Biological, Radiological, Nuclear, and Explosives (CBRNE) Sensing*, 2018.

[2] K. Bryan, M. Solomon, K. Smith, A. O. Smith, and A. M. Peter, "Deep Wavelet Scattering Features for Infrasonic Threat Identification," in *SPIE: Chemical, Biological, Radiological, Nuclear, and Explosives (CBRNE) Sensing*, 2018.

[3] M. Solomon, K. Smith, K. Bryan, A. O. Smith, and A. M. Peter, "Infrasound Threat Classification: A Statistical Comparison of Deep Learning Architectures," in *SPIE: Chemical, Biological, Radiological, Nuclear, and Explosives (CBRNE) Sensing*, 2018.

[4] N. Mijatovic, R. Haber, A. Rangarajan, A. O. Smith, and A. M. Peter, "Time Series Data Classification using Discriminative Interpolation with Sparsity," in *International Conference on Computational Science and Computational Intelligence*, 2017.

[5] S. G. Divins *et al.*, "Signal Classification Using Covariance Matrices: A Riemannian Geometry Framework," in *International Work Conference on Time Series Analysis*, 2017, pp. 400–410.

[6] R. Haber, A. Rangarajan, and A. M. Peter, "Discriminative Interpolation For Classification of Functional Data," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 9284, pp. 20–36.

[7] R. Haber *et al.*, "Functional Data Classification by Discriminative Interpolation with Features," in *International Work Conference on Time Series Analysis*, 2017, pp. 1120–1131.

[8] Y. Yan, M. Sethi, A. Rangarajan, R. R. Vatsavai, and S. Ranka, "Graph-Based Semi-Supervised Classification on Very High Resolution Remote Sensing Images," *Int. J. Big Data Intell.*, vol. 4, no. 2, pp. 108–122, 2016.

[9] S. Chakrabarti, J. Judge, T. Bongiovanni, A. Rangarajan, and S. Ranka, "Disaggregation of Remotely Sensed Soil Moisture in Heterogeneous Landscapes Using Holistic Structure-Based Models," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4629–4641, 2016.

[10] C. Yang, M. Sethi, A. Rangarajan, and S. Ranka, "Supervoxel-Based Segmentation of 3D Volumetric Images," in *ACCV*, 2016, pp. 37–53.

[11] M. A. Garces, "On Infrasound Standards, Part 1 Time, Frequency, and Energy Scaling," *InfraMatics*, vol. 02, no. 02, pp. 13–35, 2013.

[12] E. Tuncer and B. Friedlander, *Classical and Modern Direction-of-Arrival Estimation*. 2009.

[13] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Magazine*, vol. 5, no. 2. pp. 4–24, 1988.

[14] I. McCowan, "Microphone arrays: A tutorial," *Queensl. Univ. Aust.*, no. April, pp. 1–36, 2001.

[15] X. Cao, H. Liu, and S. Wu, “DOA Estimation Based on Online Music Algorithm,” *J. Electron. Inf. Technol.*, vol. 30, pp. 2658–2661, 2011.

[16] H. Kim, A. M. Haimovich, and Y. C. Eldar, “Non-coherent direction of arrival estimation from magnitude-only measurements,” *IEEE Signal Process. Lett.*, vol. 22, no. 7, pp. 925–929, 2015.

[17] W. Jiang, A. M. Haimovich, and Y. C. Eldar, “Direction of arrival by non-coherent arrays,” in *2016 IEEE InRadar Conference (RadarConf)*, 2016, pp. 1–6.

[18] E. Wigner, “On the Quantum Correction for Thermodynamic Equilibrium,” *Phys. Rev.*, vol. 40, no. 5, p. 749, 1932.

[19] L. Cohen, *Time Frequency Analysis: Theory and Applications*. Prentice Hall, 1994.

[20] J. G. Daugman, “Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters,” *J. Opt. Soc. Am.*, vol. 2, no. 7, pp. 1160–1169, Jul. 1985.

[21] M. A. Garcés, “Explosion Source Models, Chapter in Infrasound Monitoring for Atmospheric Studies,” in *Springer Lecture Notes in Computer Science*, 2017.

[22] M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation,” *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[23] G. Davis, S. Mallat, and Z. Zhang, “Adaptive time-frequency decompositions with matching pursuits,” *Opt. Eng.*, vol. 33, p. 2183.

[24] Z. Jiang, Z. Lin, and L. S. Davis, “Label Consistent K-SVD: Learning a Discriminative Dictionary for Recognition,” *IEEE Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2651–2664, 2013.

[25] G. E. Hinton, S. Osindero, and Y. W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.

[27] M. Abadi *et al.*, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015.

[28] A. Rangarajan, S. Gold, and E. Mjolsness, “A novel optimizing network architecture with applications,” *Neural Comput.*, vol. 8, no. 5, pp. 1041–1060, 1996.

[29] A. L. Yuille and A. Rangarajan, “The concave-convex procedure,” *Neural Comput.*, vol. 15, no. 4, pp. 915–936, 2003.

[30] K. Crammer and Y. Singer, “On the algorithmic implementation of multiclass kernel-based vector machines,” *J. Mach. Learn. Res.*, vol. 2, pp. 265–292, 2001.

[31] R. Tibshirani, “Regression Shrinkage and Selection Via the Lasso,” *J. R. Stat. Soc. Ser. B*, vol. 58, no. 1, pp. 267–288, 1996.

[32] S. Fischer, G. Cristóbal, and R. Redondo, “Sparse overcomplete Gabor wavelet

representation based on local competitions," *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 265–272, 2006.

[33] S. Theodoridis, *Machine Learning: A Bayesian and Optimization Perspective*, 1st ed. Academic Press, 2015.

[34] J. G. Daugman, "Neural networks for image transformation, analysis, and compression," *Neural Networks*, vol. 1, no. Supplement-1, pp. 488–491, 1988.

[35] T. S. Lee, "Image Representation Using 2D Gabor Wavelets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 10, pp. 959–971, 1996.

[36] A. Rajwade, A. Rangarajan, and A. Banerjee, "Image Denoising Using the Higher Order Singular Value Decomposition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 849–862, 2013.

[37] G. Grossi, R. Lanzarotti, and J. Lin, "Orthogonal Procrustes Analysis for Dictionary Learning in Sparse Linear Representation," *PLoS One*, vol. 12, no. 1, 2017.

[38] K. S. Gurumoorthy, A. Rajwade, A. Banerjee, and A. Rangarajan, "Beyond SVD: Sparse projections onto exemplar orthonormal bases for compact image representation," in *19th International Conference on Pattern Recognition*, 2008, pp. 1–4.

[39] K. S. Gurumoorthy, A. Rajwade, A. Banerjee, and A. Rangarajan, "A Method for Compact Image Representation Using Sparse Matrix and Tensor Projections Onto Exemplar Orthonormal Bases," *IEEE Trans. Image Process.*, vol. 19, no. 2, pp. 322–334, 2010.

[40] J. de Leeuw and K. Lange, "Sharp Quadratic Majorization in One Dimension," *Comput. Stat. Data Anal.*, vol. 53, no. 7, pp. 2471–2484, 2009.

[41] K. Lange, *Optimization*, 2nd ed. Springer, 2013.

[42] L. Zhang *et al.*, "Boosting Techniques for Physics-Based Vortex Detection," *Comput. Graph. Forum*, vol. 33, no. 1, pp. 282–293, 2014.

[43] Y. Deng, A. Rangarajan, and B. C. Vemuri, "Supervised learning for brain MR segmentation via fusion of partially labeled multiple atlases," in *13th International Symposium on Biomedical Imaging*, 2016, pp. 633–637.

[44] L. J. Cao and F. E. H. Tay, "Support vector machine with adaptive parameters in financial time series forecasting," *IEEE Trans. neural networks*, vol. 14, no. 6, pp. 1506–1518, 2003.

[45] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognit. Lett.*, vol. 42, pp. 11–24, 2014.

[46] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 85, no. 1, pp. 6–23, 1997.

[47] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *Handb. brain theory neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[48] T. Chen, A. Rangarajan, S. J. Eisenschenk, and B. C. Vemuri, “Construction of a neuroanatomical shape complex atlas from 3D MRI brain structures,” *Neuroimage*, vol. 60, no. 3, pp. 1778–1787, 2012.

[49] M. A. Garcés, “Migrating to dense, context-based sensor networks,” in *RMR Infrasound Workshop*, 2014.

[50] M. A. Garcés, “Evolution of Distributed Infrasound Sensor Networks,” in *CTBTO Infrasound Technology Workshop*, 2015.

[51] J. Schnurr, M. A. Garcés, A. Christe, and I. Y. Che, “Multiresolution analysis of transient events,” in *NNSA UITI*, 2016.

[52] F. M. Ham *et al.*, “A neurocomputing approach for monitoring plinian volcanic eruptions using infrasound,” *Procedia Comput. Sci.*, vol. 13, pp. 7–17, 2012.

[53] J. Davis and M. Goadrich, “The relationship between Precision-Recall and ROC curves,” in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 233–240.

[54] R. Baeza-Yates and B. Ribeiro-Neto, “Modern information retrieval,” *ACM Press*, vol. 9, p. 513, 1999.

[55] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[56] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[57] A. Guazzelli, M. Zeller, W. Lin, and G. Williams, “PMML: An open standard for sharing models,” *R J.*, vol. 1, no. 1, pp. 60–65, 2009.

[58] L. Richardson and S. Ruby, *RESTful Web Services*. 2008.

[59] P. Henderson and V. Ferrari, “End-to-end training of object class detectors for mean average precision,” in *InAsian Conference on Computer Vision*, 2017, pp. 198–213.

[60] M. Zhu, “Recall, precision and average precision,” *Dep. Stat. Actuar. Sci.*, pp. 1–11, 2004.

[61] S. Mika, “Kernel Fisher Discriminants,” Technical University of Berlin, 2002.

[62] A. Khamparia, D. Gupta, N. G. Nguyen, A. Khanna, B. Pandey, and P. Tiwari, “Sound classification using convolutional neural network and tensor deep stacking network,” *IEEE Access*, vol. 7, pp. 7717–7727, 2019.

8 Acronyms

AI/ML	Artificial Intelligence/Machine Learning
MASINT-X	Measurement and Signatures Intelligence Exploitation
SNR	Signal-To-Noise Ratio
FIT	Florida Institute of Technology
UoF	University of Florida
IMS	International Monitoring Systems
D3IC	Deep Discriminative Dictionaries for Infrasound Classification
SNP	Sensor Network Processor
TFDL	Time-Frequency Dictionary Learner
DFL	Deep Feature Learner
CE	Classification Engine
SVM	Support Vector Machine
DA	Directional Arrival
FFT	Fast Fourier Transform
SFFT	Short Time Fourier Transforms
OMP	Orthogonal Matching Pursuits
K-SVD	Kernel Singular Value Decomposition
API	Application Program Interface
HOSVD	Higher Order Singular Value Decomposition
LASSO	Lease Absolute Shrinkage and Selection Operator
PCA	Principal Component Analysis
NL-Means	Non Local Means
TNT	Trinitrotoluene
ROC	Receiver Operator Curve
PMML	Predictive Modeling Mark-up Language
RESTful	Representational State Transfer
MAP	Mean Average Precision
AWS	Amazon Web Services
KDA	Kernel Discriminant Analysis
LDA	Linear Discriminant Analysis
KNN	K-Nearest Neighbor
UAS	Unmanned Aircraft systems

RC	Radio Control
CNN	Convolutional Neural Network
DFFT	Discrete Fast Fourier Transform
RBF	Radial Basis Function