INL/JOU-21-61897-Revision-2

AviSense: A Real-time System for Detection, Classification, and Analysis of Aviation Signals

March 2022

Aniqua Baset, Shamik Sarkar, Sneha Kumar Kasera, Kurt W Derr, Christopher D Becker

Changing the World's Energy Future

INL/JOU-21-61897-Revision-2

AviSense: A Real-time System for Detection, Classification, and Analysis of Aviation Signals

Aniqua Baset, Shamik Sarkar, Sneha Kumar Kasera, Kurt W Derr, Christopher D Becker

March 2022

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

AviSense: A Real-time System for Detection, Classification, and Analysis of Aviation Signals

ANIQUA BASET, School of Computing, University of Utah, USA
CHRISTOPHER BECKER, Idaho National Lab, USA
KURT DERR, Idaho National Lab, USA
SHAMIK SARKAR, School of Computing, University of Utah, USA
SNEHA KASERA, School of Computing, University of Utah, USA

Wireless systems are an integral part of aviation. Apart from their apparent use in air-to-ground communication, wireless systems play a crucial role in avionic functions including navigation and landing. An interference-free wireless environment is therefore critical for the uninterrupted operation and safety of an aircraft. Hence, there is an urgency for airport facilities to acquire the capability to continuously monitor aviation frequency bands for real-time detection of interference and anomalies. To meet this critical need, we design and build AviSense, an SDR-based *real-time*, *versatile* system for monitoring aviation bands. AviSense detects and characterizes signal activities to enable practical and effective anomaly detection. We identify and tackle the challenges posed by a diverse set of critical aviation bands and technologies. We evaluate our methodology with real-world aviation signal measurements and two custom datasets of *anomalous* signals. We find that our signal classification capability achieves a true positive rate of ~99%, with few exceptions, and a false positive rate of less than 4%. We also demonstrate that AviSense can effectively distinguish between different types of anomalies. We build and evaluate a prototype implementation of AviSense that supports distributed monitoring.

CCS Concepts: • **Computing methodologies → Anomaly detection**; • **Computer systems organization → Sensors and actuators**;

Additional Key Words and Phrases: Real-time spectrum monitoring, Aviation, Spectrum anomaly detection, Autoencoders

ACM Reference Format:
Aniqua Baset, Christopher Becker, Kurt Derr, Shamik Sarkar, and Sneha Kasera. 2022. AviSense: A Real-time System for Detection, Classification, and Analysis of Aviation Signals. 1, 1 (March 2022), 34 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

This work was supported through the Idaho National Laboratory Directed Research & Development (LDRD) Program under the Department of Energy (DOE) Idaho Operations Office Contract DE-AC07-05ID14517. It was also partly supported by the National Science Foundation under Grant No. 1564287.
Authors' addresses: Aniqua Baset, aniqua@cs.utah.edu, School of Computing, University of Utah, 50 Central Campus Dr, Salt Lake City, Utah, USA, 84112; Christopher Becker, christopher.becker@inl.gov, Idaho National Lab, 1955 N Fremont Ave, Idaho Falls, Idaho, USA, 83415; Kurt Derr, kurt.derr@inl.gov, Idaho National Lab, 1955 N Fremont Ave, Idaho Falls, Idaho, USA, 83415; Shamik Sarkar, shamik.sarkar@utah.edu, School of Computing, University of Utah, 50 Central Campus Dr, Salt Lake City, Utah, USA, 84112; Sneha Kasera, kasera@cs.utah.edu, School of Computing, University of Utah, 50 Central Campus Dr, Salt Lake City, Utah, USA, 84112.

XXXX-XXXX/2022/3-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

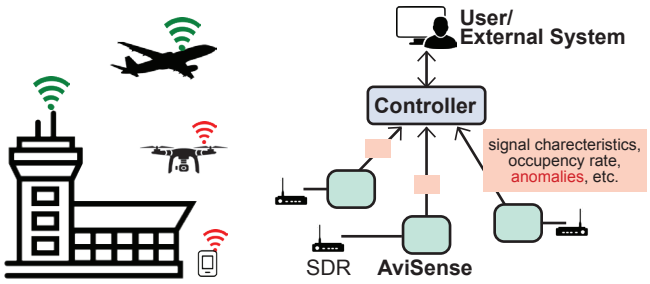
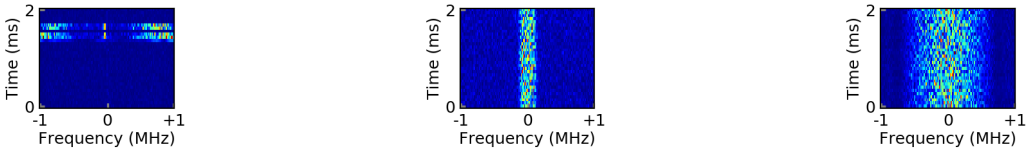


Fig. 1. AviSense for aviation spectrum monitoring.

1 INTRODUCTION

Wireless systems are an integral part of aviation. Apart from their apparent use in air-to-ground communication, wireless systems play a crucial role in avionic functions including navigation and landing. E.g., aircraft use Instrumental Landing System (ILS) signals to obtain guidance during approach and landing without visual reference [28]. Any distortion in this signal can cause an excursion with catastrophic consequences. An interference-free wireless environment is therefore critical for the uninterrupted operation and safety of an aircraft. The advent of Software Defined Radio (SDR) in recent times has multiplied the risk of interference from unauthorized use and wireless attacks, as it enables transmission over a wide range of frequencies without requiring expensive, dedicated hardware. Hence, there is an urgency for airport facilities to acquire the capability to continuously monitor aviation frequency bands for real-time detection of interference and anomalies. To meet this critical need, in this work, we design and build an SDR-based system, AviSense, for the real-time monitoring of the aviation frequency bands. We envision the use of AviSense in a distributed setting where multiple AviSense nodes placed around an airport continuously collect and send spectrum information to a controller (as depicted in Figure 1). We recognize that AviSense must meet the following objectives to be effective and of practical use in airports.

- *Anomaly detection*: AviSense must identify any deviations from the normal characteristics of aviation bands as these deviations are the indicator of situations that can affect aircraft safety and require attention.
- *Enable root cause analysis*: Interference or anomalies can arise from various sources, e.g., device malfunction, out-of-band reception, unauthorized spectrum use, malicious attacks, etc. As an example, any detected anomaly in an active ILS channel can imply a range of scenarios. The presence of an ILS signal with an unexpected frequency shift can indicate a hardware misconfiguration. The absence of an ILS signal can imply a hardware failure, in which case only background noise will be registered in that channel. However, the absence of the ILS signal in conjunction with an elevated noise floor can be indicative of an anomalous high power wide-band signal or just noise transmission. Alternately, the source of a detected anomaly can be the presence of some other signals alongside ILS. Furthermore, the presence of an ILS signal with a more than usual power level can indicate a spoofing attack [54]. Hence, just detecting an anomaly is not enough; AviSense must also facilitate root cause analysis.
- *Versatility*: AviSense must handle the unique challenges posed by the diverse set of frequency bands and technologies used in commercial aviation. E.g., the location broadcast messages from aircraft in the 1090 MHz band are short-lived (64μs to 120μs long), making these hard to detect. The 118-136.975 MHz band, which several air traffic control applications use, comprises narrow interleaved channels with bursty and sporadic transmissions. In this band, the combination of active and vacant channels varies from one airport to another resulting in dissimilar "normalcy". AviSense must systematically deal with such dissimilar normalcy for accurate classification of normal and anomalous behaviors.



(a) *Normal* (b) *Anomalous* (c) *Anomalous*

Fig. 2. Example spectrograms for 1090 MHz band.

- *Real-time*: AviSense must continuously analyze the spectrum and provide real-time spectrum information to the controller so that timely actions can be taken.
- *Adaptability*: AviSense should be usable across different compute platforms. This is important in practical scenarios, as depending on the location of deployment and budgetary constraints, aviation facilities may choose hardware with different capabilities.

We design AviSense to comprehensively achieve the above objectives. We propose a novel combination of signal detection, autoencoder (AE) based unsupervised anomaly detection, and spectrum characterization to detect and distinguish anomalies as follows. First, from incoming In-phase and Quadrature (I/Q) samples from an SDR, we distinguish the presence of signal transmission from noise. We propose a novel signal detection technique for this step that first estimates the presence of signal using a time-domain energy detector and then refines the detection decision using Spectral Flatness Measure (SFM) [30] (Section 3.5). At run-time, we periodically tune the threshold for our energy detector based on the signal detection results of SFM. Unlike existing approaches, this eliminates the need for pre-deployment efforts to estimate the necessary energy threshold and allows AviSense to adapt to changes in the noise floor. Once a signal is detected, we generate a spectrogram, i.e., the time-frequency representation of the observed spectrum, to determine whether it is *normal* for the current aviation band or not. E.g., Figure 2a shows the *normal* signal spectrogram for the 1090 MHz band, and spectrograms shown in Figures 2b, 2c are examples of *anomalous* signals. To make such a *normal-anomalous* distinction, we employ an AE-based technique customized for the aviation signals (Section 3.7.2). Our customization primarily involves the use of a shallow convolutional AE architecture. Finally, we determine the characteristics of the detected signals, such as received power, transmission frequency, occurrence rate, etc (Section 3.8). Besides these generic characterizations, we also perform aviation signal-specific measurements, e.g., the Difference in Depth of Modulation (DDM) [54] value for the ILS signal, which is crucial for detecting ILS spoofing attacks. Additionally, we keep track of the spectrum occupancy and the ambient noise level. All of this *granular spectrum information* facilitates root cause analysis at the controller. E.g., an ILS signal with a different transmission frequency than expected points to an anomaly due to hardware misconfiguration.

We design each of our above AviSense components based on our observations for a diverse set of critical aviation bands and technologies (Section 2). For instance, one of the goals of our signal detection technique is to efficiently partition continuous I/Q samples for our classification and characterization when monitoring aviation bands that have bursty and sporadic transmissions (Section 3.5). Additionally, we design AviSense as a general framework that can be configured and extended to detect and distinguish anomalies in other aviation bands not explicitly considered in this paper (Section 3.9).

We take several preemptive and reactive measures at run-time to ensure adaptability and real-timeliness. For all of our components, we devise efficient techniques to reduce computation as much as possible without relying on any computing device-specific acceleration. When our computing resources cannot keep up with the increased spectrum occupancy, we perform our signal detection step for all of the incoming I/Q samples to provide an accurate spectrum occupancy information, but leave out the computation-heavy classification of

some spectrograms, usually the ones that are most likely part of the same transmission (Section 3.6). This allows AviSense to dynamically adapt to the spectrum occupancy at run-time and operate continuously.

To evaluate AviSense, we collect real-world aviation signal data from a diverse set of locations; near a runway, inside an air traffic control tower, inside a building close-by an airport, and open space located ∼7 miles from an airport. We demonstrate that our false positive rate (FPR), that is, the fraction of *normal* signal spectrograms that get mis-classified as *anomalous*, is below 4%. We also provide an exhaustive evaluation of our *anomalous* signal detection capability and show that our true positive rate (TPR), that is, the fraction of *anomalous* signal spectrograms that an AE correctly classifies as *anomalous*, is 99%, with few exceptions. We achieve such high TPR and low FPR using our customized AEs that are 14-20x faster in comparison to generic AEs used in existing work. We also demonstrate that AviSense can effectively distinguish between different types of anomalies. Finally, we build a prototype implementation of AviSense for distributed monitoring. Notably, our prototype allows *plug-in-play* use of different SDRs, including USRP X310 [16] and HackRF One [19]. We demonstrate that our prototype can classify all of the signal spectrograms in the 108-117.95 MHz and 118-136.975 MHz band, even when the spectrum occupancy is 100%. For the bursty 1090 MHz band, our prototype only misses spectrogram classification when the spectrum occupancy goes beyond 25%. We also show that our prototype remains operational even at high spectrum occupancy.

Summary of contributions. We make the following contributions in this paper.

- We present AviSense, an SDR-based *real-time*, *versatile* system for monitoring aviation bands. AviSense detects and characterizes signal activities to enable practical and effective anomaly detection.
- We identify and tackle the challenges posed by a diverse set of critical aviation bands and technologies.
- We evaluate our methodology with real-world aviation signal measurements.
- We develop and evaluate a prototype of AviSense that supports distributed monitoring.

The rest of this paper is organized as follows. Section 2 presents an overview of current wireless technologies used in aviation and details the specific frequency bands and technologies that we consider in our work. Section 3 describes our design and methodology for AviSense. Sections 4, 5, 6 present our real-world data collection, our evaluations, and prototype implementation, respectively. Section 7 outlines the related works and finally, Section 8 concludes this paper.

2 AVIATION BANDS & TECHNOLOGIES

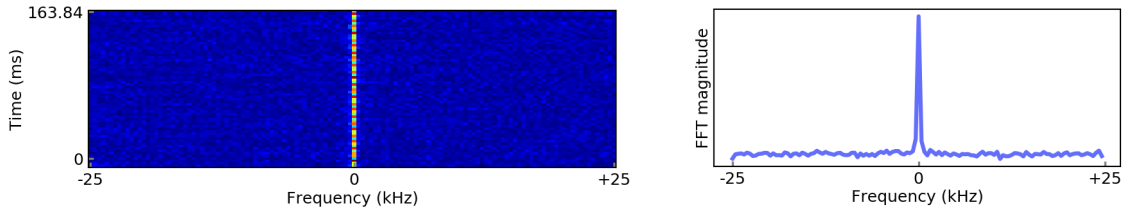
We summarize the frequency bands used in commercial aviation along with the transmission type (continuous/bursty) in each band, related aviation wireless technologies, and their applications in Table 1. Among these aviation frequency bands, in this paper, we consider the 108-117.95 MHz, 118-136.975 MHz, and 1090 MHz bands. We choose these three since they span the most significant application areas of aviation including air traffic control, information services, and navigational aid [60], and two very different types of signal transmissions—continuous and bursty. Below, we provide an overview of these bands and their associated technologies.

2.1 108-117.95 MHz (VHF-108)

This band contains 50 kHz interleaved channels of Instrument Landing System-Localizer (ILS-LOC) and VHF Omnidirectional Radio Range (VOR). ILS-LOC provides navigational guidance to an aircraft during approach and landing, and VOR helps an aircraft determine its bearing and stay on course during the flight [28]. Both ILS-LOC and VOR have constant transmissions. The ILS-Glideslope signal, part of the 328.6-335.4 MHz band, has the same working principle as ILS-LOC. Our work applies to both ILS-LOC and ILS-Glideslope. Thus, in the rest of this paper, we generally use ILS to refer to both of these. We present example spectrogram and FFT of ILS and VOR in Figures 3 and 4, respectively.

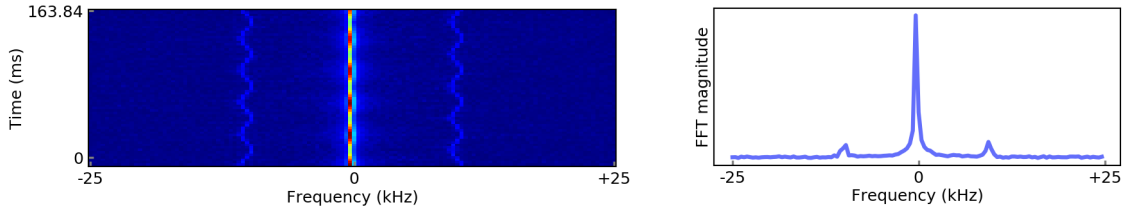
Table 1. List of frequency bands and corresponding wireless technologies used in commercial aviation. The names in the parenthesis are the abbreviations used in this paper. We focus on the first three frequency bands in this work.

Frequency band	**Transmission type**	**Technology**	**Application**
108-117.95 MHz (VHF-108)	Continuous	Instrument Landing System (ILS) – Localizer	Navigational aids
		VHF Omnidirectional Radio Range (VOR)	Navigational aids
118-136.975 MHz (VHF-118)	Bursty & sporadic	Aircraft Communications Addressing and Reporting System (ACARS)	Information services
		Controller–Pilot Data Link Communications (CPDLC)	Air traffic control
		Voice	Air traffic control
1090 MHz (L-1090)	Bursty & sporadic	Automatic Dependent Surveillance-Broadcast (ADS-B)	Air traffic control
		Multilateration (MLAT)	Air traffic control
		Secondary Surveillance Radar (SSR) – Reply (Mode A/C/S)	Air traffic control
		Traffic Alert and Collision Avoidance System (TCAS)	Information services
		Traffic Information System-Broadcast (TIS-B)	Information services
190–1750 kHz	Continuous	Non-directional Beacon (NDB)	Navigational aids
328.6-335.4 MHz	Continuous	ILS – Glideslope	Navigational aids
978 MHz	Bursty & sporadic	ADS-B	Air traffic control
		TIS-B	Information services
		Flight Information System-Broadcast (FIS-B)	Information services
962–1213 MHz	Bursty	Distance-measuring Equipment (DME)	Navigational aids
1030 MHz	Fixed rate	Secondary Surveillance Radar (SSR) – Interrogation	Air traffic control
1.2276 & 1.57542 GHz	Continuous	Global Positioning System (GPS)	Navigational aids
2.7-2.9 GHz	Fixed rate	Primary Surveillance Radar (PSR)	Air traffic control



(a) Spectrogram (b) FFT

Fig. 3. Example of ILS.



(a) Spectrogram (b) FFT

Fig. 4. Example of VOR.

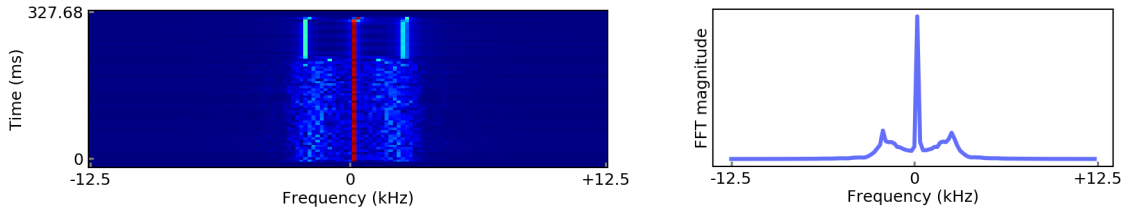
2.2 118-136.975 MHz (VHF-118)

This band comprises 25 kHz channels that are used for air traffic control and information services. Unlike VHF-108, VHF-118 channels have bursty and sporadic transmissions. Among the VHF-118 aviation technologies, we consider Aircraft Communications Addressing and Reporting System (ACARS) in this paper. Our methodology for ACARS applies to other VHF-118 technologies as well. ACARS provides a digital communication link between aircraft and ground stations to exchange air traffic control messages, flight phases, engine status, etc. ACARS uses Amplitude Modulation—Minimum Shift Keying and its transmission duration can be 170–910 ms long [3]. Figure 5 shows example spectrogram and FFT of ACARS.

2.3 1090 MHz (L-1090)

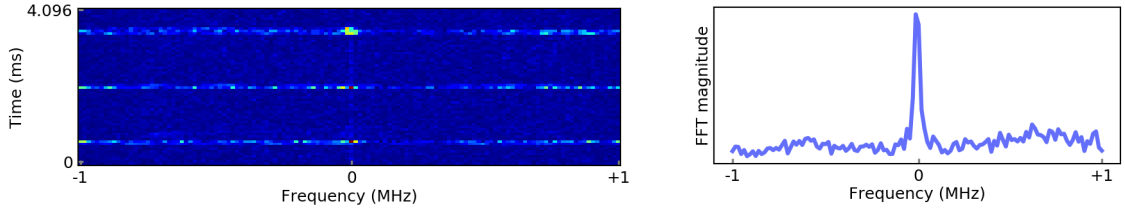
This band is used by a number of aviation technologies as depicted in Table 1. It is mainly used for Automatic Dependent Surveillance-Broadcast (ADS-B) [50] and for transmitting replies from aircraft in response to Secondary Surveillance Radar (SSR) interrogations [39]. Other technologies, such as, MLAT, TCAS, and TIS-B are the indirect users of this band as they do not have separate data links and utilize the transmitted information intended for ADS-B and SSR. All of the technologies of this band play a critical role in obtaining the location and velocity of every aircraft, and thus is essential for air traffic control and collision avoidance.

ADS-B and SSR replies have similar physical layer characteristics, and we refer to them as L-1090, in this paper. L-1090 signals use pulse position modulation (PPM) with 64–120 μs long duration. There can be a frequency shift of up to ±1 MHz in the received L-1090 signal. Therefore, we consider a 2 MHz band centered at 1090 MHz as L-1090 in this work. The transmissions in this band are bursty and sporadic. Figure 6 shows example spectrogram and FFT of L-1090.



(a) Spectrogram with a single transmission. (b) FFT

Fig. 5. Example of ACARS.



(a) Spectrogram with three transmissions. (b) FFT

Fig. 6. Example of L-1090.

3 AVISENSE DESIGN

In this section, we describe AviSense in detail. We begin by defining the terms that we repeatedly use in this paper in Section 3.1. Next, we present AviSense's system architecture in Section 3.2, followed by our analysis of different types of anomalies and their detection by AviSense in Section 3.3. Then, we detail the key components of AviSense in Sections 3.4—3.7. Finally, we describe our aviation band-specific configurations in Section 3.9 and summarize AviSense in Section 3.10.

3.1 Definitions

Signal & noise. We refer to a set of I/Q samples or any representation of these (e.g., Fast Fourier Transform (FFT), spectrogram) that contains signal transmission as *signal*. Conversely, we use *noise* to refer to the I/Q sample set that contains just the ambient noise.

Active & inactive channel. An *active* channel is the one that is being used at an airport. E.g., JFK airport uses the 109.5 MHz, 110.9 MHz, 111.35 MHz, and 111.5 MHz ILS-LOC channels [4]. Therefore, these channels are *active* at/near JFK, and the rest of the 40 ILS-LOC channels are *inactive*.

***Normal* & *anomalous* signal.** The expected signal in a monitored channel is considered *normal*, e.g., ILS centered at 110.9 MHz is considered *normal* in the VHF-108 channel. Any other signals present in this channel are considered *anomalous*. However, the presence of a *normal* signal can be an anomaly depending on the context. For instance, the presence of the ILS signal in an inactive channel is an anomaly.

Spectrogram & classification window. A spectrogram is a set of contiguous FFTs. To generate a spectrogram, we first take a bin-wise average of the magnitudes of n_a FFTs, where each of the FFTs is of size n_f. Then, we stack n_r such averages as depicted in Figure 7. Therefore, we use $n_r * n_a * n_f = n_t$ I/Q samples. We use averaging of the FFT magnitudes to reduce the effects of random noise. We refer to each of the n_r averages as a *row* throughout this section. We refer to the n_t I/Q samples needed to construct one spectrogram as our classification window, T_c.

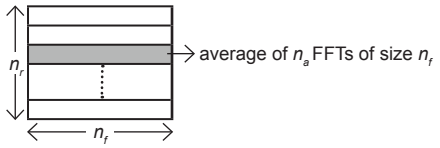


Fig. 7. Spectrogram.

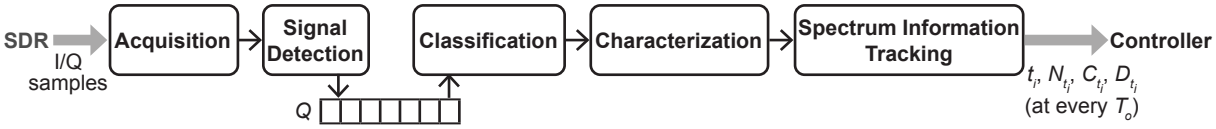


Fig. 8. System architecture of AviSense.

3.2 System Architecture

Our main intuition in designing AviSense is that different anomalies including those due to hardware malfunction, hardware misconfiguration, unauthorized spectrum use, and wireless attacks, will each have a different impact on the characteristics of the observed spectrum. Therefore, extracting granular spectrum information can maximize the probability of anomaly detection and facilitate root cause analysis. Figure 8 depicts our AviSense system architecture, which consists of five modules. The first module, Acquisition, continuously receives the I/Q samples from an associated SDR and forwards those to the Signal Detection module. The Signal Detection, Classification, Characterization, and Spectrum Information Tracking modules collectively extract actionable spectrum information to detect anomalies and facilitate root cause analysis. Signal Detection module discerns the presence of *signal* versus *noise*. It disregards the *noise* samples, and generates and forwards a curated set of *signal* spectrograms for further analysis. Classification module classifies the *signal* spectrograms as either *normal* or *anomalous*. Subsequently, Characterization module determines the detected signals' characteristics including center frequency, bandwidth, and received power. Finally, Spectrum Information Tracking module communicates with the other modules and aggregates the spectrum information that needs to be sent to the controller. Specifically, this module tracks and sends the following spectrum information to the controller periodically every T_o time units.

- *Timestamp, t_i:* It is the end of the current output period.
- *Channel information, C_{t_i}:* This includes general characteristics of the monitoring channel—the occupancy rate, r_o, and the current noise level, p_{noise}. p_{noise} is the average noise level over $[t_i - T_o, t_i]$. We define r_o as the percentage of the *signal* samples among the total samples received from the SDR over $[t_i - T_o, t_i]$. r_o is particularly important for monitoring a constant channel like ILS/VOR. For an active ILS/VOR channel, r_o should always be 100%. Encountering a lower value for r_o is therefore, a cause for concern. Additionally, a sudden increase in p_{noise} can indicate anomalous activities in the monitored or a nearby channel.
- *Node information, N_{t_i}:* This contains information specific to an AviSense node—the current location, l_{t_i}, and the current signal classification rate, r_c. We define r_c as the percentage of the spectrograms that the node is able to classify among the total *signal* spectrograms detected over $[t_i - T_o, t_i]$. r_c is an indicator of the overall classification throughput of the node. Therefore, one can choose the computing platform for AviSense based on the desired or tolerated r_c.
- *Detected signal characteristics, D_{t_i}:* This provides the characteristics of the signals detected over $[t_i - T_o, t_i]$. For each detected signal, AviSense records signal class (i.e., *normal* or *anomalous*), s_c; center frequency, s_f; bandwidth, s_b; classification score, s_s; received power level, s_p; and occurrence rate, s_r. s_r is the percentage

of the spectrograms that are classified as s_c among the total spectrograms that we process over $[t_i - T_o, t_i]$. Besides these generic information, D_{t_i} also includes aviation signal-specific parameters, e.g., Depth of Modulation (DDM) for ILS signals.

T_o is a configurable parameter for AviSense. It can be set to a value ranging from subseconds to minutes. However, a short T_o might not provide any useful insight on the signal activities, and a long T_o can delay any investigative action at the controller. As shown in Figure 8, we introduce a fixed-length queue (Q) to temporarily hold the spectrograms that need to be transferred from Signal Detection module to Classification module. Q is beneficial when the classification of the spectrograms cannot keep up with their generation. In such instances, without Q, the associated SDR will report "overflows" and become non-operational, hampering the continuous operation of AviSense.

3.3 Detecting Different Types of Anomalies

Anomalies are usually categorized into three types—point, contextual, and collective [11]. Towards meeting our goal of spectrum anomaly detection in aviation bands, we define these three anomaly types as follows:

- **Point anomalies:** These anomalies can be detected based on just the current spectrogram at run-time. The presence of an *anomalous* signal is an example of such an anomaly. For constant-signal channels of VHF-108, the presence of frequency-shifted *normal* signals is also a point anomaly.
- **Contextual anomalies:** For these anomalies, we need to take into account the context including AviSense's location, the measurement channel, historical data, and the time of the measurement. For example, ILS is a directional signal, and is not expected to be observed at all parts of an airport, e.g., ILS must not be present at terminals. The detection of ILS signals at an unexpected location is a contextual anomaly with the context being AviSense's location. In a different scenario, consider the normal occurrence rate of L-1090 signals after midnight, based on a month's observation at a fixed location, to be between s_{r_min} and s_{r_max}. Encountering an occurrence rate higher than s_{r_max} after midnight at the same location is a contextual anomaly.
- **Collective anomalies:** In this case, the spectrum information of a single output period might appear to be normal, but can be anomalous when examined collectively over multiple output periods.

We design AviSense to detect point anomalies and to collect spectrum information to facilitate the detection of contextual and collective anomalies. Different reasons for anomalies, such as, hardware malfunction, out-of-band interference, spoofing attacks, and jamming, will result in different types of anomalies. For example, a jamming attack that transmits non-aviation signals will cause a point anomaly that can be directly detected by AviSense. On the other hand, if a jamming attack uses aviation signals, AviSense will not identify that as an anomaly. However, such a jamming attack will likely change the occupancy rate, signal occurrence rate, or received power. In that case, AviSense's spectrum information can be used to detect the attack. We summarize different types of anomalies in aviation bands and their detection by AviSense in Table 2.

3.4 Acquisition Module

This module acts as an interface for acquiring the digitized signal data from an associated SDR. It continuously receives I/Q samples, that is, the complex discrete time-domain signal from the SDR, and forwards those to the subsequent Signal Detection module. Depending on the type of the SDR used, the Acquisition module can incorporate additional processing of I/Q samples before forwarding. For example, the I/Q samples from HackRF One SDRs contain DC bias. This bias results in high energy at the center frequency in the FFT [65], which is misleading for signal detection and classification tasks. Therefore, we incorporate a *DC offset removal* step in the implementation of our Acquisition module for HackRF One following [22, 44]. At the start of the process of acquiring data from an HackRF One SDR, our module computes the averages of I samples and averages of Q

Table 2. Different anomalies and their detection by AviSense

Anomaly	Cause	Type	Detection	When missed
Anomalous signals	Any type of jamming attack that is using non-aviation signal, unauthorized spectrum use, or out-of-band interference	Point	Directly by AviSense	AviSense cannot classify the signal as *anomalous* ‡
Frequency-shifted *normal* signals in VHF-108	Hardware misconfiguration	Point	Directly by AviSense	Frequency shift is less than AviSense's frequency resolution
Absence of *normal* signals in VHF-108	Hardware failure	Contextual	Based on AviSense's spectrum information and location context	
Constant *normal* signals in VHF-118 or L-1090	Constant jamming attack	Contextual	Based on AviSense's spectrum information and context	
High power *normal* signals	Overshadow attack	Contextual	Based on AviSense's spectrum information and context	Power is very close to historical normal values.
Spoofed signals overriding part of ILS signals	Spoofing attack	Contextual	Based on AviSense's spectrum information and context	DDM is very close to historical normal values
Spoofed *normal* signals	Random, reactive, or deceptive jamming attack that transmits *normal* signals or spoofing attack	Contextual, Collective	Based on AviSense's spectrum information and context	Attacks change neither the received power, occupancy rate, or occurrence rate

‡ This might occur due to low r_{bw}, r_p, or r_d as analyzed in Section 5.1.3

samples for a few seconds and does not forward any sample during this time. Later, it subtracts the computed averages from each I and Q sample before forwarding those to the Signal Detection module.

3.5 Signal Detection Module

For this module, we need a signal detection technique that does not require prior knowledge about the signal's characteristics. Additionally, we require this technique to be computationally inexpensive to keep up with the I/Q sample rate without dropping any sample. Dropping I/Q samples at this stage will result in missed signal/noise classification. The most common signal detection technique, energy-based thresholding [69], fits these criteria. However, if we use energy-based thresholding, a *noise* spectrogram with elevated energy will get forwarded to

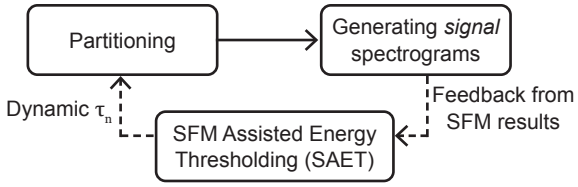


Fig. 9. Components of Signal Detection module.

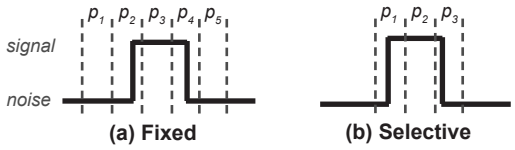


Fig. 10. Partitioning. Here, p_i = one partition.

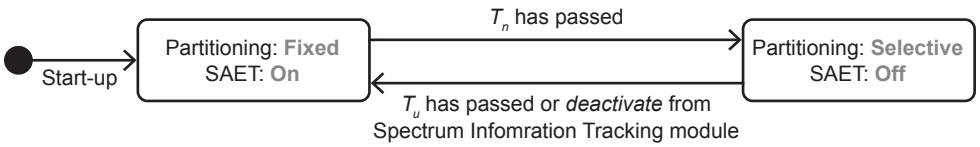


Fig. 11. Partitioning & SAET.

Classification and classified as *anomalous*. It will then appear that there is an *anomalous* in-band signal. Instead, we aim to classify such a spectrogram as *noise* with elevated energy. Then, this will indicate an anomaly due to noise transmission or wide-band signal transmission with a flat spectrum in our monitored channel. We propose to use the spectral flatness measure (SFM) based signal detection to efficiently and accurately classify spectrograms as *signal* or *noise* without any prior knowledge of the signals. SFM is the ratio of the geometric mean to the arithmetic mean of the FFT magnitudes [30]. An SFM value close to 1 indicates noise. Following [6], we consider SFM > 0.96 as *noise* and SFM ≤ 0.96 as *signal*.

At run-time, the following components jointly perform signal detection and generate the corresponding spectrogram, S, for the subsequent modules (Figure 9).

3.5.1 Partitioning. We perform an initial partitioning of the incoming continuous I/Q samples to potentially generate a row for S. Hence, a partition corresponds to a set of $n_a * n_f$ consecutive I/Q samples. We consider two approaches for creating a partition, fixed and selective. In our *fixed* approach, we continuously start partitioning from the end of the last partition. For *selective*, we apply insight from the I/Q samples before starting a partition as follows. We first divide the samples into blocks of size n_b, compute their received signal strength indicator (RSSI), and start the partition when there are n_c consecutive blocks with RSSI greater than an energy threshold, τ_n. We propose *selective* partitioning for the channels where bursty and sporadic transmissions are expected for the following reasons. First, we can get fewer partitions than those from *fixed* (Figure 10), hence less SFM computation. Second, *selective* reduces the chances of misclassifications due to partial information, especially for the short-lived L-1090 signals.

3.5.2 SFM Assisted Energy Thresholding (SAET). Assuming the availability of a static τ_n that would be determined based on pre-deployment measurements for every aviation band, location (indoor/outdoor), and SDR hardware is not practical. We get rid of such pre-deployment efforts by devising an automated method, SFM Assisted Energy Thresholding (SAET). SAET determines τ_n at run-time based on the *signal-noise* decision of SFM. It computes τ_n as the median of RSSIs of the blocks corresponding to the partitions that SFM classified as *noise* over a period T_n. Some blocks might contain the start or end of a signal and thus can mislead τ_n computation. Our use of the the median removes the effect of such outliers. We repeat the computation of τ_n at random intervals, T_u. When we activate SAET to compute τ_n, we turn off *selective* partitioning. Additionally, to avoid missing signals due

to incorrect τ_n, SIT module also turns off the *selective* partitioning when r_o drops below a specified threshold. Figure 11 shows the how Signal Detection module transitions between fixed and selective partitioning, and when SAET is turned off/on.

3.5.3 Generating signal spectrograms. We compute a spectrogram row from a partition and label it as *signal* or *noise* based on its SFM value. Once we find a *signal* row, we add that to S and wait for the next partitions for the needed n_r rows to complete S. If we find a *noise* row, we add that to S if S is not empty; otherwise, we just ignore that. Once S is full, we perform a *partial* spectrogram detection step. If the number of *signal* rows in S is less than τ_{sr}, we mark S as *partial* and directly forward it to Characterization module; otherwise, forward it to Q. τ_{sr} is a configurable parameter and can be adjusted based on the *normal* signal characteristics of the aviation channel being monitored. We ignore classifying spectrograms marked as *partial* because they contain insufficient *signal* information and thereby, also save computation. The controller can use *partial* spectrograms for potential root cause analysis. For instance, presence of *partial* spectrograms in constant channels like ILS/VOR can be construed as *anomalous* requiring further investigation. D_{t_i}

3.6 Managing Q

As stated in Section 3.2, Q temporarily holds the spectrograms that need to be transferred from Signal Detection module to Classification module. We clear Q at every T_o after sending our spectrum information to the controller.

3.6.1 Choosing Q length, Q_l. When r_c <100%, Q_l spectrograms among all of the *signal* spectrograms generated by Signal Detection module over time T_o get forwarded to Classification module. If Q_l is very large, only spectrograms that appear at the beginning of T_o period will get forwarded, but the rest will be missed. Therefore, we need to set Q_l somewhat smaller so that we can introduce randomness in classification and in the same time not miss any periodic signal that are always present towards the end of T_o. We set Q_l based on r_c for a particular device using a dummy I/Q sample file with r_o = 100%.

3.6.2 Replacement strategy for Q. For a bursty transmission that spans more than n_t samples, we will get a series of *signal* spectrograms all belonging to the same transmission. Classifying one spectrogram from such a series is enough. Therefore, when Q is full, we replace a spectrogram belonging to a series with the newly arriving spectrogram. In absence of such series, we just pick a random Q entry to replace. When a spectrogram series represents back-to-back different types of signals, our approach becomes similar to replacing random entries without utilizing any spectrogram series information.

3.7 Classification Module

This module classifies a received *signal* spectrogram from Q as *normal* or *anomalous*. For this classification, we need an unsupervised anomaly detection technique that does not need label *anomalous* signal data since it is not feasible to create a dataset of all possible *anomalous* signals. There is a multitude of unsupervised anomaly detection methods to choose from [9, 11, 43]. In this paper, we opt for the Autoencoder (AE) based anomaly detection approach for the following reasons. AE is the most widely used neural-network based anomaly detection method [9, 43]. It has been adopted in a diverse range of application domains [2, 5, 10, 12, 51, 73, 74] including spectrum anomaly detection [18, 35, 47]. We utilize AE differently than prior spectrum anomaly detection studies as we describe in Section 3.7.2. Below, we present a brief overview of alternate anomaly detection methods categorized according to [11] and discuss their applicability to our system.

The *classification based anomaly detection* methods learn a classifier using the normal training data and use the classifier to decide whether an unseen instance is normal or anomalous [11]. This category of methods can be further subdivided into one-class and multi-class groups. Any one-class classification based anomaly detection technique, e.g., One-Class Support Vector Machine (OC-SVM) [48], Support Vector Data Description (SVDD)

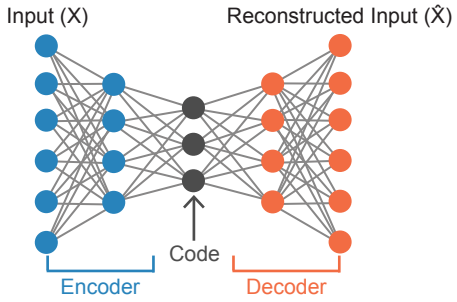


Fig. 12. Autoencoder (AE).

[63], Autoencoder (AE) [21], is relevant to our problem. Among the different options in this category, we choose AE because of its widespread use, as explained earlier. The *spectral anomaly detection* techniques assume that given data can be projected onto a different subspace where anomalies are significantly different from normal data and find such subspace to identify anomalies [11]. One of the well-known spectral methods is Principal Component Analysis (PCA) based anomaly detection [31]. While the standard linear PCA-based method is not suitable for high-dimensional data like ours [51], the kernel PCA based anomaly detection is a viable option [26]. However, as evaluated in [51], AE is a better choice over kernel PCA in terms of computational complexity. The *nearest neighbor based anomaly detection* techniques determine if a data instance is anomalous based on its distances to nearest neighbors or the density of its neighborhood [11]. Notable techniques in this category include Local Outlier Factor (LOF) [8], Connectivity-based Outlier Factor (COF) [62], and Angle-Based Outlier Detection (ABOD) [33]. Nearest neighbor based anomaly detection methods are computationally expensive since a test instance needs to be compared with all training data to find nearest neighbors. Hence, these methods are not suitable for our real-time system. The *clustering based anomaly detection* methods are based on the assumption that anomalies are farther away from their closest cluster center or they belong to sparse or small clusters [11]. Choosing a suitable clustering technique and distance measure is a challenging task. In contrast, employing an AE is more practical since it can learn the necessary features from the data itself. The *statistical anomaly detection* methods fit a statistical model to training data and use statistical inference test on a test instance to decide whether it is normal or anomalous [11]. Statistical methods are not suitable for problems like ours that involve high-dimensional data, mainly for two reasons. First, the assumption that data are generated from a particular distribution does not often hold for high-dimensional data. Next, determining a suitable statistic for high-dimensional data is challenging. The *information theoretic anomaly detection* methods expect that anomalies introduce irregularities in the information content of a dataset [11]. To identify anomalies, these methods search for a subset of data such that removing that subset reduces the irregularities in the information content of the given dataset. Information theoretic anomaly detection methods are appropriate for non-real-time applications or naturally ordered data, e.g., sequential and spatial data. Therefore, these methods do not apply to our real-time *anomalous* signal detection problem.

3.7.1 Primer on AE. AE is a class of neural networks that is trained to reconstruct its input [21]. AE comprises an encoder and a decoder network as illustrated in Figure 12. The encoder transforms the input, $\boldsymbol{X}$, into a latent representation, *code*, and the decoder reconstructs the input from the *code* producing output, $\hat{\boldsymbol{X}}$. The idea for the AE-based anomaly detection is that an AE trained on only normal data will not be able to reconstruct an anomalous input accurately. Therefore, it will have a higher error in reconstructing an anomaly, that is, in producing $\hat{\boldsymbol{X}}$ from $\boldsymbol{X}$. Hence, using a threshold for reconstruction error makes it possible to distinguish between

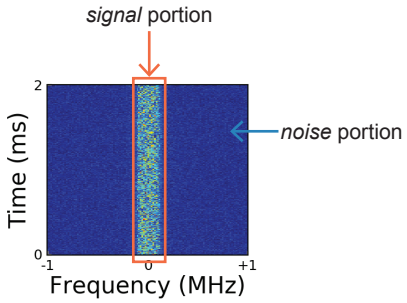


Fig. 13. *Signal* versus *noise* portions of a spectrogram.

normal and anomalous data. Generally, the reconstruction error is represented as the mean squared error (MSE) between $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$.

3.7.2 Customizing AE-based anomaly for normal-anomalous signal classification. To enable root cause analysis, we train an AE differently than existing spectrum anomaly detection studies. After data collection for an aviation signal s_a, we incorporate the following two preprocessing steps. First, we introduce synthetic frequency shifts in our training spectrograms so that we can classify any frequency-shifted s_a signal as *normal* instead of *anomalous*. Without such data augmentation during training, such a signal will be classified as *anomalous* and misguide the root cause analysis at the controller. Second, we use power-normalized spectrograms in our training so that the AE does not get affected by individual FFT magnitudes, rather it learns the pattern in the input.

We utilize a different mechanism to determine the reconstruction error than the conventional MSE. MSE treats all of the errors of a reconstructed spectrogram equally. However, for our *anomalous* signal detection task, errors in reconstructing the *signal* portions of an input spectrogram are more critical than those of the *noise* portions. For example, reconstruction errors in the area bounded by the rectangle in Figure 13 should contribute more than the rest in discerning an *anomalous* signal from a *normal* signal. Therefore, we use the following weighted mean square error (WMSE) as our reconstruction error between $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$ to assign more weights to errors corresponding to the *signal* portions of an input spectrogram.

$$\text{WMSE}\ (\boldsymbol{X}, \hat{\boldsymbol{X}}) = \frac{1}{n} \frac{\sum_{i=1}^{n} \boldsymbol{X}_i (\boldsymbol{X}_i - \hat{\boldsymbol{X}}_i)^2}{\sum_{i=1}^{n} \boldsymbol{X}_i} \tag{1}$$

As we demonstrate in Section 5.1.5, our WMSE-based approach provides higher value of Area Under the ROC Curve (AUC) than MSE.

3.7.3 Training phase. For every supported aviation signal s_a, we create a training dataset following our pre-processing steps presented in Section 3.7.2. We then train AE_{s_a} to be used in our Classification module. Next, we compute WMSE-based reconstruction errors on the training data from AE_{s_a}. Finally, we set the anomaly threshold, denoted τ_{s_a}, as the pth percentile value of the training reconstruction errors. This corresponds to considering (100-p)% of the training data as *anomalous* or tolerating (100-p)% FPR in classification. p can be set to a high percentile value depending on our confidence on the quality of the training data for s_a. In this paper, we use p to be 98th percentile (Section 5.1).

3.7.4 Run-time workflow. We first normalize a spectrogram received from Q and feed it to a pre-trained AE_{s_a}. Next, we compute the reconstruction error as WMSE and classify the spectrogram as *normal* if the computed reconstruction error is smaller than or equal to τ_{s_a}, otherwise classify it as *anomalous*. Finally, we incorporate a

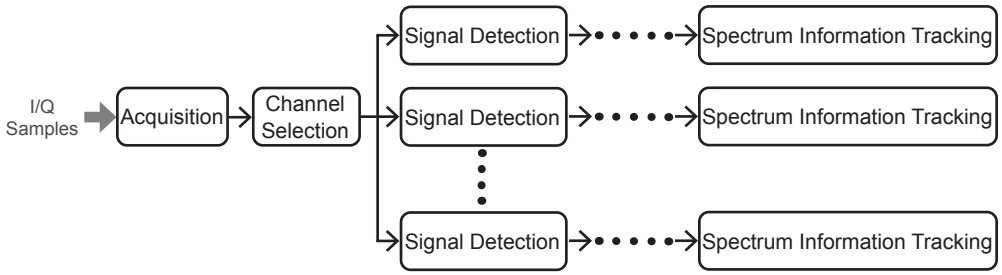


Fig. 14. Multi-channel mode for VHF-108 & VHF-118.

classification confidence score in the [0,1] range based on the computed reconstruction error and τ_{s_a}. The further the reconstruction error is from τ_{s_a}, the higher the score would be. The controller can choose to ignore the results with low scores, that is, the ones that have reconstruction errors very close to τ_{s_a}.

3.8 Characterization Module

This module determines the characteristics of a detected signal that would be useful for further analysis at the controller. We identify that, at a minimum, we need the capability to estimate the received power (s_p), the center frequency of a *normal* signal (k_c), and the center frequency and bandwidth of an *anomalous* signal (u_c and u_b). s_p helps detect anomalous power changes in *normal* signals. k_c is useful to detect any anomalous frequency shift in constant signals like ILS/VOR. u_c and u_b helps understand whether or not all detected *anomalous* signals are present at a fixed frequency or multiple frequencies and do all of these have same or different bandwidth. Such insight will allow the controller to investigate the source of *anomalous* signals. We estimate s_p as the RSSIs of the I/Q samples corresponding to a *signal* spectrogram. We estimate k_c and u_c, u_b from the bin-wise average of a spectrogram (S_{avg}) as follows.

3.8.1 Estimating k_c. This estimation depends on the power spectrum characteristics of the detected *normal* signal. We find that, for ILS, VOR, ACRS, and L-1090, estimating k_c as the maximum value index of S_{avg} is sufficient. However, this approach might not always work for other aviation signal classes. So, we also devise a general method for estimating k_c for additional aviation signals in AviSense. Since we are dealing with a *normal* signal, we know the expected bandwidth, and we utilize that for estimating k_c. Our methodology is as follows: we sequentially search for the set of FFT bins of width equal to the bandwidth of the detected signal class with the maximum sum in S_{avg} and then consider the center of such a set as k_c.

3.8.2 Estimating u_c, u_b. We can employ any peak detection method [15, 23] for these but using a complex one will not be feasible for our real-time system. Therefore, we devise the following approach inspired by the work on power thresholding-based peak detection [49]. From S_{avg}, we determine a threshold as the p% of the maximum value of S_{avg}. Next, we look for contiguous sets of bins that are above that threshold. We then compute the u_c and u_b from the center and width of the detected sets. For low signal-to-noise ratio (SNR) cases, such thresholding might split a single signal power spectrum into multiple segments. To handle this, we merge neighboring sets if their distance is less than an allowed threshold. However, if the sets are further apart, we consider them separate signals and estimate u_c and u_b for each of them.

3.9 Frequency Band Specific Configurations

3.9.1 Multi-channel bands, VHF-108 & VHF-118. The 10 MHz VHF-108 band has 50 kHz ILS and VOR interleaved channels. For monitoring VHF-108, if we consider the entire 10 MHz as a whole, we will just get a binary normal

Algorithm 1 DDM estimation for ILS

procedure DDM(F, f)
  $b_{90} \leftarrow 90/f$ ▷ # of FFT bins corresponding to 90Hz
  $b_{150} \leftarrow 150/f$ ▷ # of FFT bins corresponding to 150Hz
  $c_m, c_i \leftarrow$ maximum value and corresponding index of F
  $DM_{90} \leftarrow$ (Average of of $F[c_i - b_{90}]$ & $F[c_i + b_{90}]$)/c_m
  $DM_{150} \leftarrow$ (Average of $F[c_i - b_{150}]$ & $F[c_i + b_{150}]$)/c_m
  return DM_{90} - DM_{150}

versus anomaly decision, and not the granular spectrum information that we expect to obtain in AviSense to enable root cause analysis. Furthermore, we will need to train different AEs for different locations since not all of the channels are going to be active in all locations, and different channel types (active/inactive) will create completely different *normal* scenarios. A similar argument will hold for the VHF-118 band that is composed of 25 kHz channels. Additionally, the transmissions in the VHF-118 channels are bursty. Therefore, there will be more variations in the *normal* scenarios in VHF-118 compared to VHF-108. Hence, we include a multi-channel mode in AviSense that allows monitoring individual VHF-108/VHF-118 channels closely, if needed. For this mode, we consider a specific channel as our monitored unit, that is, a 50kHz VHF-108 channel or a 25kHz VHF-118 channel. We set the SDR to capture the whole 10 MHz VHF-108/VHF-118 band, and we now employ a slightly enhanced architecture as depicted in Figure 14 to monitor the band. Our key enhancement, in comparison to the general architecture of Figure 8, is the introduction of a Channel Selection module right after Acquisition. This Channel Selection module first selects n_p channels to monitor and sends the data corresponding to the selected channels to n_p parallel processing paths similar to the general architecture of Figure 8 (from Signal Detection module onward). It changes the assigned channels of each path after every T_o. The value of n_p would be specific to the computing platform being used for AviSense. Choosing a large n_p would require more parallel paths and hence reduce the signal classification rate, r_c. A suitable value for n_p can be determined based on the target r_c and the capability of the computing platform. The controller can set this channel selection module to only select from a list of active and critical channels of an airport or it can assign higher weights to these channels while allowing other channels.

3.9.2 Short-lived L-1090 signals. L-1090 signals have a very short duration; the maximum duration being 120 µs. Therefore, averaging n_a FFTs might lead to missing or misclassifying L-1090 signals. To address this issue, we use $n_a = 1$ for L-1090. We also make the following adjustments to our Signal Detection module workflow (Section 3.5) for L-1090. We use a partition of size $n_r * n_f$ to generate a spectrogram. Our SFM-based signal detection decision is for the whole spectrogram instead of just for a row.

3.9.3 Constant channels, ILS & VOR. For constant ILS & VOR signals, there is no additional information or pattern available with respect to time. We find that an average power spectrum is enough as the input to the AEs of ILS and VOR instead of spectrograms. Therefore, for ILS and VOR, we use spectrograms with $n_r = 1$. We deactivate *selective* partitioning (Section 3.5) for ILS & VOR since it is not applicable when such constant transmission is expected.

3.9.4 Detecting spoofing attack against ILS. The ILS-LOC transmitters transmit amplitude modulated signals with 150 Hz and 90 Hz tones such that these tones dominate the right and left side of the runway centerline (RCL), respectively. The difference in depth of modulation (DDM) of an ILS signal is defined as the difference in the relative amplitudes of 150 Hz and 90 Hz tones with respect to its carrier's amplitude. DDM guides an aircraft to align with RCL. An aircraft receiver will experience DDM = 0 at RCL, DDM < 0 on the right side of RCL, and DDM > 0 on the left side of RCL. As demonstrated in [54], an attacker can transmit spoofed ILS signals with

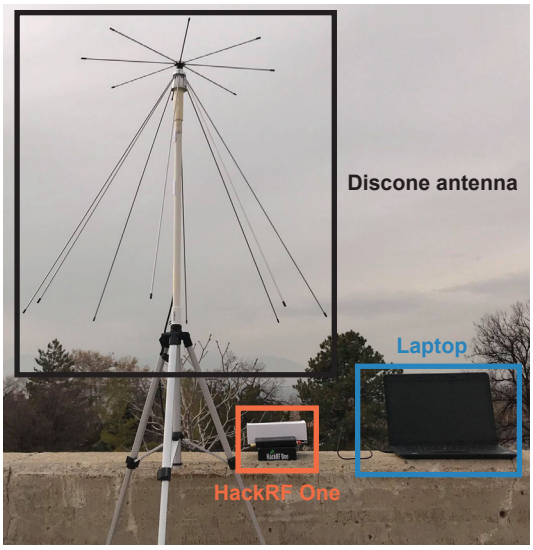


Fig. 15. Our hardware setup for data collection.

modified DDM to misguide aircraft during landing. Therefore, spoofing attacks against ILS can be detected by monitoring the changes in the DDM at a fixed receiver location. Hence, we incorporate a DDM estimation for ILS in our Characterization module for spoofing detection. Our methodology to estimate DDM from a bin-wise average of a set of FFTs (F) and the frequency resolution (f) is presented in Algorithm 1. Another motivation for adding DDM to Characterization module is to demonstrate that such aviation signal-specific capability can be integrated into AviSense to broaden its usability.

3.10 Summary

AviSense provides two options for monitoring an aviation band—general and multi-channel mode. The general mode is for bands like L-1090 that are being used by a single type of aviation signal. The multi-channel mode is for bands like VHF-108/VHF-118 consisting of interleaved narrowband channels of different aviation signals. This mode allows users to closely monitor the critical aviation channels that are active at an airport. Before deploying AviSense, for every supported aviation signal s_a, we need to follow the steps outlined in Section 3.7 to train AE_{s_a} for our Classification module. The training is one-time and location-agnostic, i.e., every AviSense node employs the same AE_{s_a}. Our general mode can also be used to monitor the whole VHF-108/VHF-118 band. However, as described earlier, we will get a binary decision (normal/anomaly) by doing so, not the granular spectrum information we aim to provide with AviSense. Additionally, we will need location-specific AEs for the bands. In summary, AviSense provides a comprehensive framework that is configurable to monitor the aviation frequency bands with different granularity as needed. We customize the functionalities of AviSense modules to meet the aviation signal-specific needs.

4 DATASETS

4.1 Aviation signal data collection

Figure 15 illustrates our data collection setup. We use an SDR (USRP X310/HackRF One/RTL-SDR), a Tram 1410 Broad Band Discone antenna, and a laptop as our receiver. We run a GNU Radio [20] based script on the laptop

Table 3. Summary of our aviation signal data collection

Location	Description	Frequency bands	Day(s)	SDR used
Near-Runway	Close to the runway and just outside of airport A	VHF-108	Multiple	HackRF One
Inside-Airport	Air Traffic Control Tower (ATCT) of airport B	VHF-108, VHF-118, L-1090	Single	USRP X310
Building-Near-Airport	Inside 2nd floor of a building located ~1 mile from airport A	L-1090	Single	RTL-SDR
Outside-Airport	Elevated, open space located ~7 miles from airport B	VHF-118, L-1090	Multiple	HackRF One

Table 4. Number of spectrograms in *normal* datasets

Signal	Training	Test
ILS	80,538	725,310
VOR	25,220	37,838
ACARS	110,686	135,288
L-1090	93,309	740,806

to capture and save I/Q samples in a file. We refer to such saved files as *recordings* throughout the rest of this paper. We collect recordings for aviation frequency bands VHF-108, VHF-118, and L-1090 at various locations, as summarized in Table 3. We collect data at different locations and on different days to investigate the robustness of our classification.

4.2 Generating *normal* training and test datasets

We first filter out the active channel signal data from our wide-band recordings. Since ILS and VOR channels have continuous transmissions, we use all the spectrograms of these channels from the recordings. For bursty channels, ACARS and L-1090, we identify the *normal* transmissions as follows. For identifying *normal* L-1090 transmissions, we use a modified version of dump1090 [52], a software-defined L-1090 receiver. For ACARS, we compute the noise level threshold based on the SFM decisions (Section 3.5). Next, we label the continuous I/Q sample sets as ACARS signals whose RSSIs are above our threshold and lengths conform with the ACARS specification.

For generating spectrograms, we set $n_f = 128$ for all signals, $n_a = 4$ for ACARS, and n_a, n_r values for rest of the aviation signal types as described in Section 3.9. We then follow our preprocessing steps of Section 3.7 to create our datasets of spectrograms. Table 4 presents the numbers of spectrograms in our final datasets. We include more variations in our test dataset in comparison to our training dataset in terms of the frequency shifts, percentage of the *signal* rows in a spectrogram, the number of recordings, the SDR used in data collection, gain setting of the SDR, and collection day/location/time. Our purpose for incorporating such variations is to investigate the

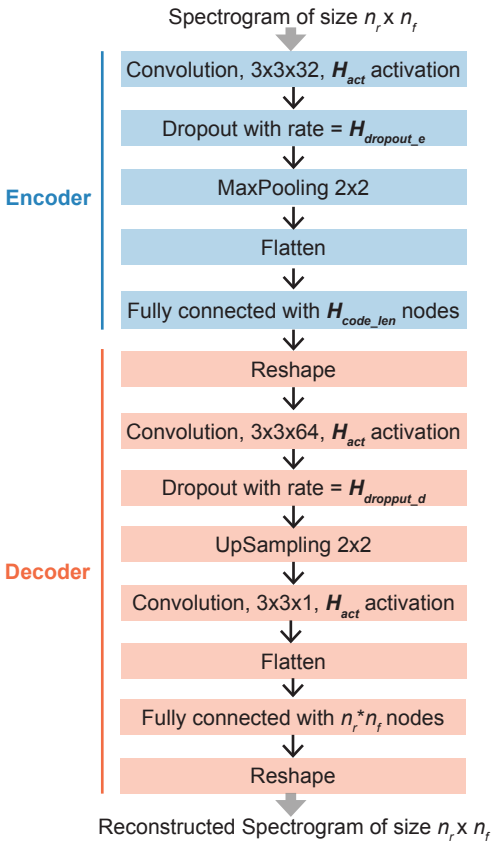


Fig. 16. Our AE architecture for $n_r > 1$. For $n_r = 1$, we use one-dimensional convolution with filter size 3 and one-dimensional Max-Pooling with pool size 2. We choose the values for H_{act}, $H_{dropout\_e}$, $H_{dropout\_d}$, H_{code_len} via 5-fold cross-validation. We skip the dropout layers if cross-validation finds $H_{dropout\_e} = 0$ or $H_{dropout\_d} = 0$.

practicality and effectiveness of deploying AviSense with pre-trained AEs at different locations where it can encounter varied *normal* spectrograms than those used in training.

4.3 Generating *anomalous* test datasets for evaluation

We utilize the following two datasets as our examples for *anomalous* signals in our evaluation.

(1) Anomalous-Anechoic: We generate random signals with different modulation techniques and bandwidths with GNURadio, and transmit in an anechoic chamber. We use a similar receiver setup, as described in Section 4.1, to capture these signals. We next create spectrograms from the captured data. We vary the signal-to-noise ratio (SNR) of these spectrograms by adding random noise.
(2) Anomalous-RadioML: We use the dataset from [40]. It contains signal traces of both simulated and over-the-air recordings of 24 modulation techniques. The recordings also have different SNR levels.

We have a total of 1M and 3M spectrograms in Anomalous-Anechoic and Anomalous-RadioML, respectively.

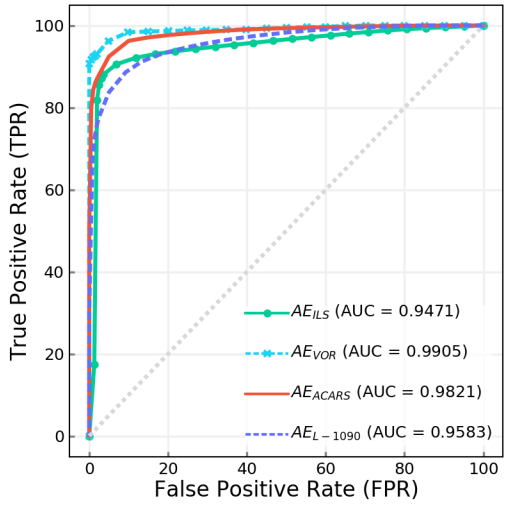


Fig. 17. Overall performance of *normal-anomalous* signal classification.

5 EVALUATION

5.1 Evaluating *normal-anomalous* signal classification

In this section, we evaluate our AE-based *normal-anomalous* signal classification. We begin by presenting our process for training the AEs in Section 5.1.1. Next, we present our overall classification performance in Section 5.1.2. Then, we provide an exhaustive analysis of our *anomalous* signal detection performance for a diverse set of *anomalous* signal scenarios in Section 5.1.3, followed by our *normal* signal detection results in Section 5.1.4. Finally, we demonstrate the advantage of our WMSE-based reconstruction error approach in Section 5.1.5 and compare the throughput of our trained AEs with that of generic spectrum anomaly detection in Section 5.1.6. In all of these evaluations, we utilize the *normal*, *anomalous* test datasets that we described in Sections 4.2 and 4.3, respectively.

5.1.1 Training. We train AE_{ILS}, AE_{VOR}, AE_{ACARS}, and AE_{L-1090} on the training dataset that we presented in Section 4.2 using Keras [13] with a Tensorflow [1] backend. We adapt the convolution AE architecture from [46] for our AEs. We find that a shallower network compared to [46] is sufficient for the aviation signals. Adding more convolution layers in the encoder and decoder networks increases the prediction time without significantly improving the classification performance. Figure 16 illustrates our adapted AE architecture. Our encoder network consists of a convolution layer followed by a dropout layer with $H_{dropout\_e}$ rate, a max-pooling layer, and a fully connected layer with H_{code_len} nodes. Our decoder network comprises a convolution layer followed by a dropout layer with $H_{dropout\_d}$ rate, an upsampling layer with nearest neighbor interpolation, a second convolution layer, and a fully connected layer with $n_r * n_f$ nodes. The activation functions used in our convolution layers and fully connected layers are H_{act} and $linear$ activation, respectively. For each of our AEs, we choose the hyperparameters, that is, H_{act}, $H_{dropout\_e}$, $H_{dropout\_d}$, H_{code_len}, and the optimizer [21], using a 5-fold cross-validation [24]. We skip the dropout layers if our cross-validation step finds $H_{dropout\_e} = 0$ or $H_{dropout\_d} = 0$.

5.1.2 Overall performance. We use the Receiver Operating Characteristic (ROC) curve and the Area Under the ROC Curve (AUC) to represent our overall classification performance. Figure 17 shows the ROC curves and the corresponding AUC values of our AEs. Here, true positive rate (TPR) represents the fraction of *anomalous* signal

spectrograms that an AE correctly classifies as *anomalous* and false positive rate (FPR) represents the fraction of *normal* signal spectrograms that get mis-classified as *anomalous*. We also note in Figure 17 that all of our AEs result in high AUC values.

5.1.3 Anomalous signal detection. Here, we provide a detailed evaluation of our TPR, that is, how accurately our AEs can classify input spectrograms with *anomalous* signals as *anomalous*. For the results presented in this section, we use the 98th percentile value of the training reconstruction error as our anomaly threshold, which corresponds to 2% FPR in training data.

Depending on the channel that we are monitoring and the comparative duty cycle of *normal* and *anomalous* signals in that channel, the following three scenarios are possible.

(A) Presence of just *anomalous* signal in an active channel: For constant ILS/VOR channels, this scenario can only occur when an *anomalous* signal has high transmission power and it completely overrides the constant *normal* signal. In contrast, for bursty ACARS and L-1090, this happens when an *anomalous* signal is present during the idle period of the *normal* signals. For evaluating this scenario, we use our Anomalous-Anechoic and Anomalous-RadioML datasets. We find that the TPR mainly depends on *anomalous* signal's bandwidth with respect to the monitored bandwidth. Therefore, we present our TPR with respect to the ratio of the *anomalous* signal bandwidth to the monitored bandwidth, r_{bw}, in Figures 18a, 18b, 18c, 18d. Our results show that we get ∼99% TPR for *anomalous* signals whose $r_{bw} \geq$20% for all of our AEs. For lower r_{bw}, the TPR varies depending on the target *normal* signals. For instance, AE_{VOR} can detect *anomalous* signals with ∼ 99% TPR even when their r_{bw} is as low as ∼5%. From our results, we can conclude that AviSense will only miss-classify a very small subset of *anomalous* signals—signals bandwidth smaller than 0.2*2 MHz = 400 kHz when monitoring L-1090 or 0.1*50 kHz = 5 kHz when monitoring ILS.

Figure 18e presents our TPR with different SNR levels for AE_{L-1090}. Our result shows that we get high TPR for signals with SNR > 5 dB when $r_{bw} \geq 10$. For signals with narrower bandwidths, the performance varies depending on the *anomalous* signal type. We observe similar results for rest of the AEs.

(B) Simultaneous presence of *anomalous* signal and *normal* signal in an active channel: This is a more plausible scenario for constant ILS/VOR channels than (A). For evaluating this scenario, we combine the spectrograms from our *normal* test dataset of Table 5 and the spectrograms of a QAM modulated signal whose r_{bw}=20%, as an *anomalous* signal example from Anomalous-Anechoic. To include a diverse combination of *normal* and *anomalous* signals, in our test spectrograms, we vary the ratio of the *anomalous* signal's peak versus the *normal* signal's peak (r_p), frequency overlap between *normal* and *anomalous* signals, and the ratio r_d of the *anomalous* signal duration to classification window, T_c. Figure 19 shows that spectrograms with higher r_p and r_d have higher TPR. In general, spectrograms with no frequency overlap have higher TPR than the ones with frequency overlap. These results match with our intuition that a higher r_p, higher r_d, and/or no frequency overlap corresponds to an AE failing to reconstruct a bigger portion of a test spectrogram. For low r_p and low r_d cases, AE_{L-1090} is not as accurate as the other AEs due to the following reason. Normally, there are a large number of variations in L-1090 data compared to other signals. For instance, there can be multiple short-lived L-1090 transmissions in a single spectrogram separated by time and/or frequency, which result in different patterns than a single transmission. Therefore, it is challenging for an AE to learn the *normal* pattern resulting in a high variance in the training reconstruction error. This contributes to the low detection accuracy for low r_p and low r_d cases for AE_{L-1090}.

(C) Presence of *anomalous* signal in an inactive channel: Detecting an *anomalous* signal in this scenario is essentially the same as (A). However, we are at an advantage here since the presence of *anomalous* signals will increase the r_o where we actually expect $r_o = 0$. Therefore, based on r_o, we can identify an anomaly even if an *anomalous* signal spectrogram gets classified as *normal*.

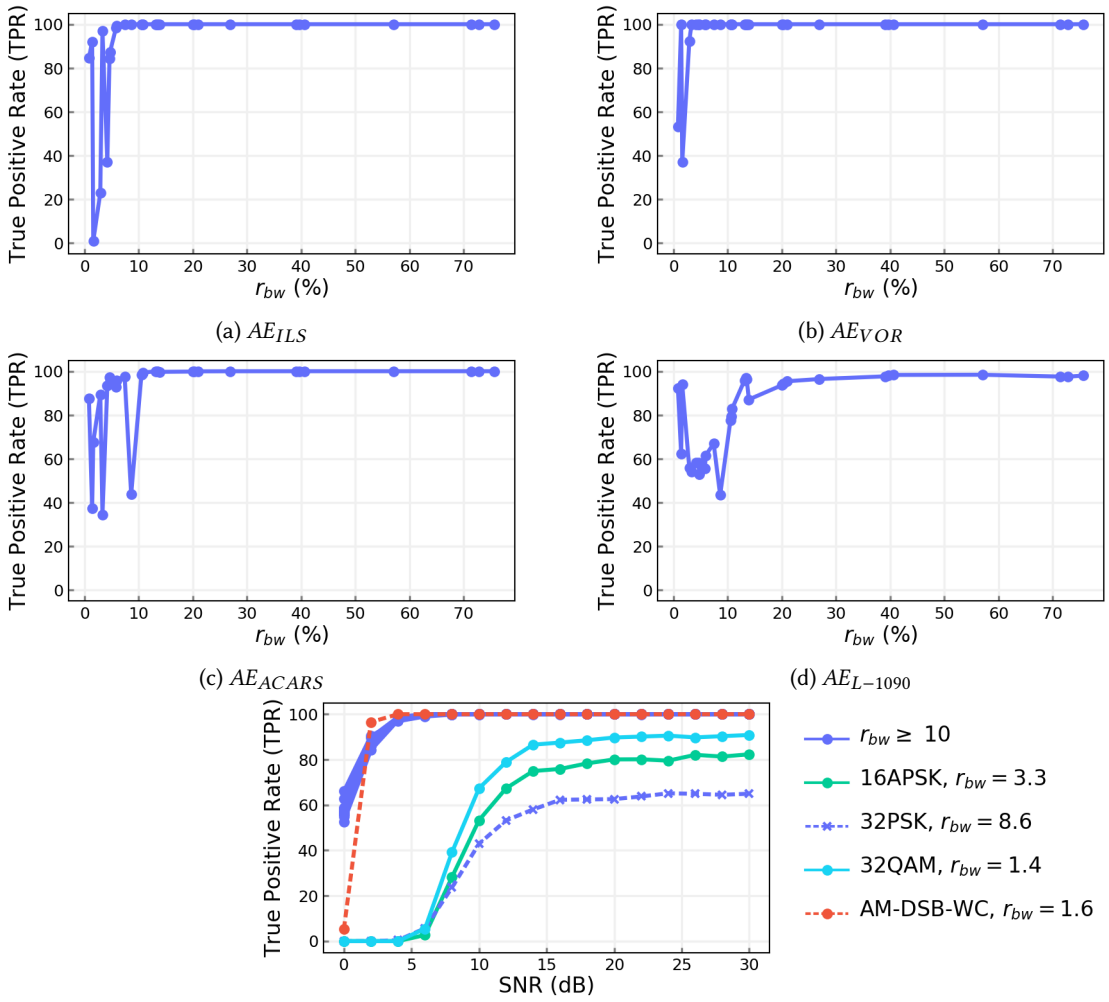
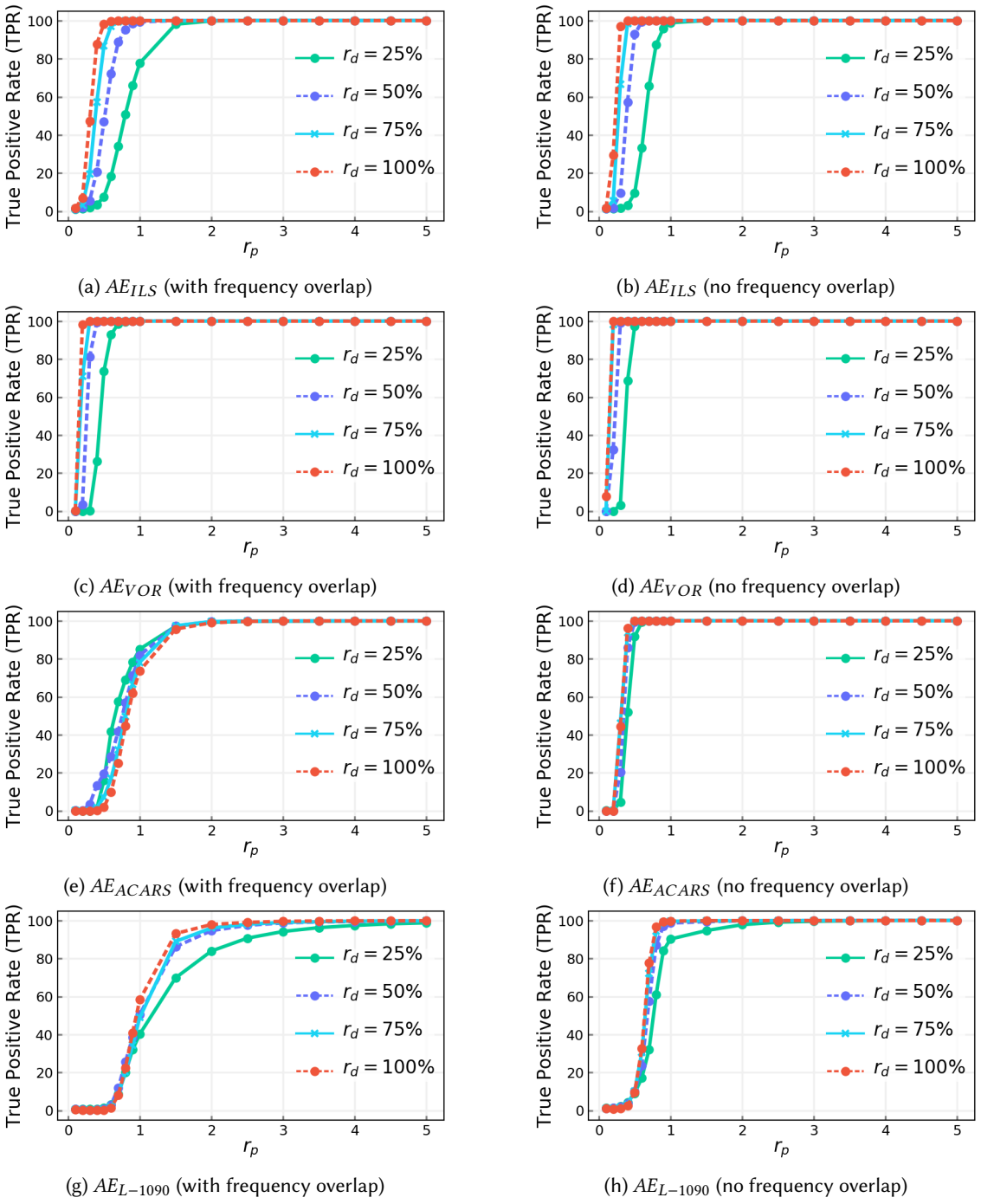


Fig. 18. TPR when just *anomalous* signal is present. Here, r_{bw} = ratio of the *anomalous* signal's bandwidth to the monitored bandwidth.

5.1.4 Normal signal detection. Our FPRs[1] of AE_{ILS}, AE_{VOR}, AE_{ACARS}, and AE_{L-1090} on our *normal* test dataset are 3.62%, 1.98%, 2.03%, and 2.36%, respectively. The low FPRs, i.e., the high detection accuracy of *unseen normal* signal spectrograms, *different* from those used in training, demonstrates the robustness of our trained AEs and also the practicality of deploying pre-trained AEs at different locations.

5.1.5 Advantage of using WMSE over MSE. As we described in Section 3.7.2, we use WMSE (Equation 1) for computing the reconstruction error as opposed to conventional approach of using MSE. The ROC curves in

[1]As in Section 5.1.3, we present our results corresponding to the anomaly threshold set to the 98th percentile value of the training reconstruction error.



(a) AE_{ILS} (with frequency overlap)

(b) AE_{ILS} (no frequency overlap)

(c) AE_{VOR} (with frequency overlap)

(d) AE_{VOR} (no frequency overlap)

(e) AE_{ACARS} (with frequency overlap)

(f) AE_{ACARS} (no frequency overlap)

(g) AE_{L-1090} (with frequency overlap)

(h) AE_{L-1090} (no frequency overlap)

Fig. 19. TPR when both *anomalous* and *normal* signals are present. Here, r_p = ratio of the *anomalous* signal's peak to the *normal* signal's peak, and r_d = ratio of the *anomalous* signal's duration to classification window, T_c.

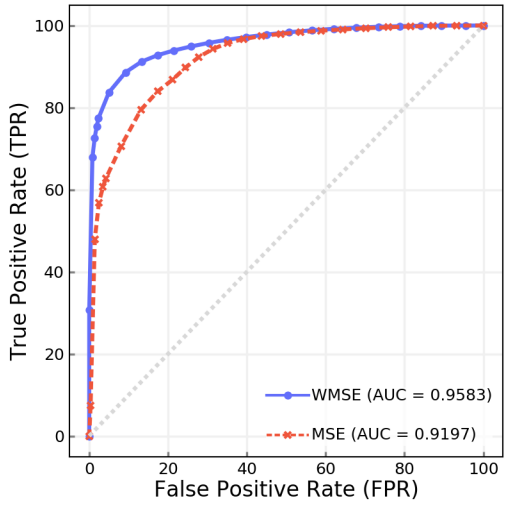


Fig. 20. Effect of using WMSE versus MSE based reconstruction error for AE_{L-1090}

.

Table 5. Throughput comparison

Method	**L-1090**	**ACARS**
AE (this work)	1787.77	1252.4
AE from [46]	87.35	88.84

Figure 20 demonstrate the overall performance gain in using WMSE over MSE for AE_{L-1090}. We observe similar improvement in rest of our AEs.

We also observe improvement in the FPR when using WMSE. For instance, the FPR[1] of AE_{L-1090} decreases from 4.25% to 2.36% when we use WMSE instead of MSE.

5.1.6 Comparison with generic spectrum anomaly detection. We compare the throughput of our resultant AEs with that of the state-of the-art AE-based generic spectrum anomaly detection [46]. We define throughput as the number of predictions that an AE can perform per second. Table 5 shows that our AEs are 14-20x faster than [46].

5.2 Evaluating SAET

Here, we evaluate our SAET mechanism for dynamic computation of τ_n of our Signal Detection module. For this evaluation, we utilize several L-1090 recordings from Building-Near-Airport and Outside-Airport. We do not know the ground truth of our real-world measurements, so we consider the decision from SFM as the ground truth since SFM has 99% accuracy for most signals with SNR $\geq$ 5 dB and for some signals with SNR $\geq$ 0 dB as we demonstrated in [6]. We evaluate τ_n in terms of False Negative Rate (FNR) and False Positive Rate (FPR). FNR represents the percentage of *signal* partitions that τ_n misses, and FPR represents the percentage of *noise* partitions that τ_n misclassifies as *signal*. Table 6 presents our τ_n computation, test setup, and results. Table 6 shows that we get very low FNR, i.e., τ_n misses very few signals. In general, for data from the same day as τ_n computation, we get FPR = ~33-50%. This implies that *selective* partitioning eliminates 50-67% *noise* partitions, hence reduces the

Table 6. Evaluating SAET with n_b = 32 I/Q samples, n_t = 1024 I/Q samples, and T_n = 0.5 min

Data used in computing τ_n	Test setup		Result	
	Data	Test period	FNR (%)	FPR (%)
Building-Near-Airport	Same day as computation	119.5 min	0.01	32.79
Outside-Airport	Same day as computation	19 min	0.41	49.54
	Different day than computation	20 min	0.0	74.20

computation needed for SFM. However, we get high FPR, i.e., less advantage of using *selective* partitioning when our test data is from a different day than our τ_n computation. This demonstrates that it is not practical to use a static τ_n continuously; hence we need our dynamic and automated τ_n computation mechanism, SAET.

5.3 Evaluating DDM estimation for ILS signals

ILS is critical for aircraft landing, and spoofing attacks against it can have catastrophic consequences [54]. As demonstrated by Sathaye *et al.*, ILS spoofing attacks will either affect both received signal power and DDM values or just DDM values [54]. Therefore, our signal power level, s_p and DDM values which are part of D_{t_i} will aid in detecting spoofing attacks. Since DDM is essential for both types of ILS attacks, we mainly focus on evaluating our DDM estimation method (Section 3.9.4) in this section.

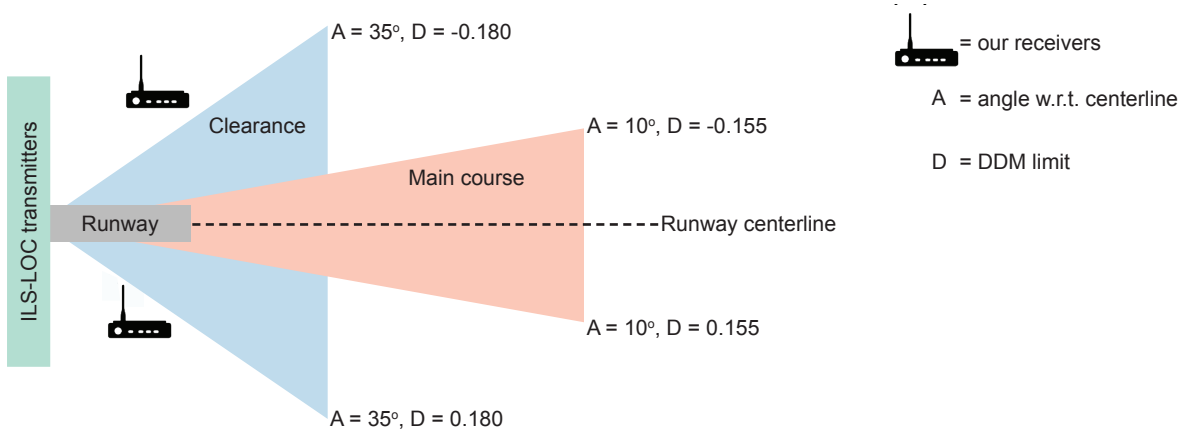
The DDM values vary from location to location depending on the relative position from the runway centerline. Therefore, to evaluate our DDM computation, we collect data at two different locations with respect to the centerline of the runway at Near-Runway as shown in Figure 21a. In the figure, the main course is the coverage area of the ILS-LOC transmitters, and clearance is the coverage area of clearance signals transmitted to override the effects of the side-lobe of ILS-LOC. There is a linear relationship between the angle and the DDM values inside the main course. However, such a relationship is not guaranteed outside of this region. There are limits on the DDM values on the boundary of the main course and clearance, which are ±0.155 and ±0.180, respectively. As described in Section 3.9, DDM = 0 at the runway centerline, DDM < 0 on the right side of the centerline, and DDM > 0 on the left side of the centerline when facing the ILS-LOC transmitters. We present out estimated DDMs for Near-Runway recordings in Figure 21b. Figure 21b shows that we get opposite DDM values for the opposite sides of the runway as expected. Our values also reflect that our receiver locations are outside of clearance.

5.4 Use Case: Differentiating Anomalies

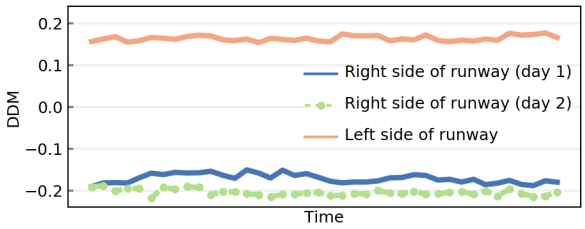
Here we evaluate the benefits of our granular spectrum information (N_{t_i}, C_{t_i}, D_{t_i}) in differentiating anomalies. For this evaluation, we utilize two of our 20 minutes long ILS recordings from Near-Runway as representatives of the normal environment, Normal-I & Normal-II. We summarize the AviSense results for Normal-I & Normal-II in Table 7a. Next, we modify Normal-II using GNURadio blocks to create different anomaly scenarios. Below, we describe these scenarios and the corresponding results that we get from AviSense.

First, we consider a hardware misconfiguration scenario that causes an anomalous frequency shift in the ILS signal, more than the regular carrier frequency offset (A-I). We add a 15 kHz frequency shift in Normal-II to create our target anomaly. Table 7b shows that we can identify the presence of shifted ILS signal from our AE_{ILS} result and the center frequency value of D_{t_i}.

Second, we consider hardware malfunction that results in absence of continuous ILS signal (A-II). For this, we set AviSense to monitor an inactive ILS channel of Normal-II as active. Our Signal Detection module does not detect any signal, so it does not forward anything to the Classification module. Therefore, as Table 7b shows, we



(a) Our data collection setup



(b) DDM estimation results

Fig. 21. Evaluating DDM at Near-Runway.

get occupancy rate, r_o = 0 % in C_{t_i}, whereas the expected r_o is 100%. Additionally, the ambient noise level of C_{t_i} shows that it is lower than the normal ILS signal power of Normal-I & Normal-II, -34.90 dB versus -30.93 dB.

Third, we consider an unauthorized wide-band transmission scenario that completely overrides the existing ILS signal and has a flat spectrum in the 50 kHz ILS channel (A-III). To create such a scenario, we combine Normal-II with high power random noise that overrides an active ILS channel. Similar to A-II, the Signal Detection module only detects noise. Table 7b shows that we get a similar effect on r_o as the previous hardware malfunction case of A-II. However, we get higher noise power than the normal ILS signal power level, which indicates additional transmission in the channel.

Last, we consider the presence of an *anomalous* signal along with ILS (A-IV). Table 7b shows that we get an *anomalous* decision from the AE, which is different from the rest of the anomaly scenarios we consider so far.

Our above evaluation shows that the granular spectrum information provided by AviSense facilitates differentiation of anomalies.

5.5 Use Case: Detecting Contextual Anomalies

We now consider a contextual anomaly scenario for L-1090 and describe its detection using AviSense. L-1090 is vulnerable to *ground station target ghost injection attack* where malicious entities transmit L-1090 signals to create ghost aircraft [37]. This attack misleads the surveillance system at ground stations. Such an injection attack is impossible to detect at AviSense's classification module since the injected signals are *normal* L-1090 signals. However, the attack will change the occupancy rate, one aspect of the spectrum information that AviSense collects.

Table 7. Evaluating the usefulness of $N_{t_i}, C_{t_i}, D_{t_i}$ for distinguishing anomalies

(a) Normal scenarios

Scenarios	**Decision from** AE_{ILS}	r_o **in** C_{t_i} **(%)**	**Noise level in** C_{t_i}**(dB)**	***normal* signal parameters in** D_{t_i}	
				RSSI (dB)	**Center frequency (MHz)**
Normal-I	*normal*	100	-	-30.86	111.1000
Normal-II	*normal*	100	-	-30.93	111.1000

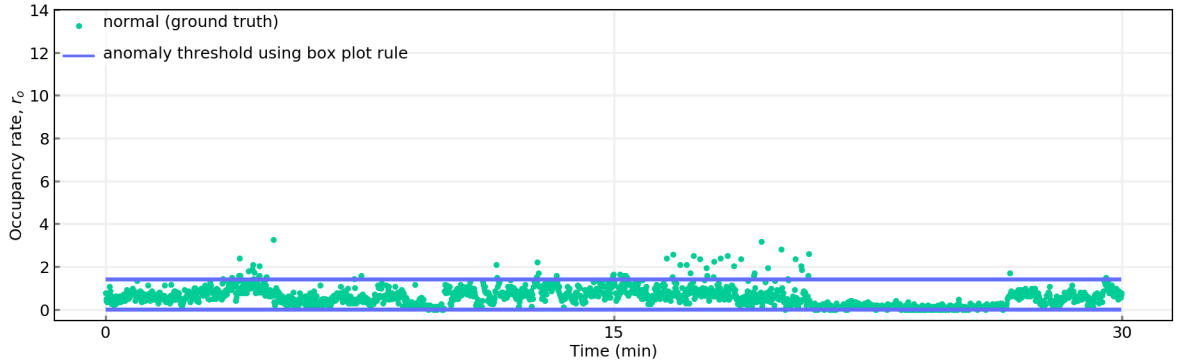
(b) Anomaly scenarios. The shaded cells represent the information affected by the anomaly present.

Scenarios	**Decision from** AE_{ILS}	r_o **in** C_{t_i} **(%)**	**Noise level in** C_{t_i}**(dB)**	***normal* signal parameters in** D_{t_i}	
				RSSI (dB)	**Center frequency (MHz)**
A-I: ILS signal with 15 kHz frequency shift	*normal*	100	-	-30.93	111.1148 MHz
A-II: Absence of expected ILS signal	-	0	-34.90	-	-
A-III: High power wide-band transmission with a flat spectrum in the ILS channel	-	0	-4.9	-	-
A-IV: Presence of *anomalous* signal along with ILS	*anomalous*	100	-	-	-

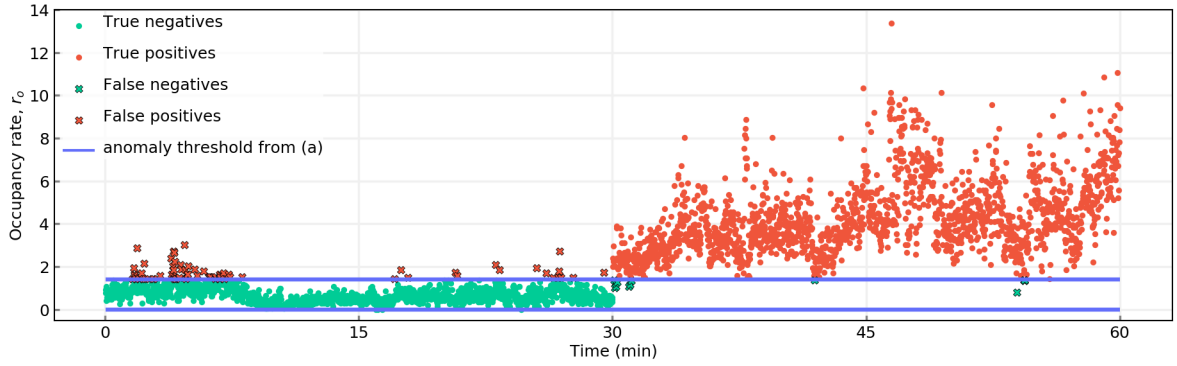
This would allow the controller to detect the attack based on the history of occupancy rates at fixed locations reported by AviSense. This attack is an example of contextual anomalies where we cannot make the anomaly decision just from current spectrograms and need to consider the location and past spectrum information.

We employ a statistical anomaly detection approach using the *box plot rule* [11] to baseline the occupancy rates at Building-Near-Airport from a 30 min recording. Figure 22a shows the normal occupancy rates at Building-Near-Airport and our *box plot rule* based thresholds, τ_{bpr}. We simulate an injection attack as follows. We append a 30 min recording at Outside-Airport to a different 30 min recording at Building-Near-Airport. Being close to an international airport and outdoors, Outside-Airport recordings have more L-1090 transmissions than indoor Building-Near-Airport recordings. This creates an attack scenario that starts after 30 min and continues for another 30 min. We achieve 97.6% accuracy in detecting anomalies in this simulated attack recording using our anomaly thresholds, τ_{bpr}. Figure 22b details our anomaly detection result.

This simple anomaly example demonstrates the benefit of the spectrum information that AviSense collects, in detecting contextual anomalies.



(a) Normal occupancy rates at Building-Near-Airport.



(b) Occupancy rates at Building-Near-Airport under a simulated injection attack that starts at 30 min.

Fig. 22. Evaluating detection of L-1090 injection attack using occupancy rates reported by AviSense.

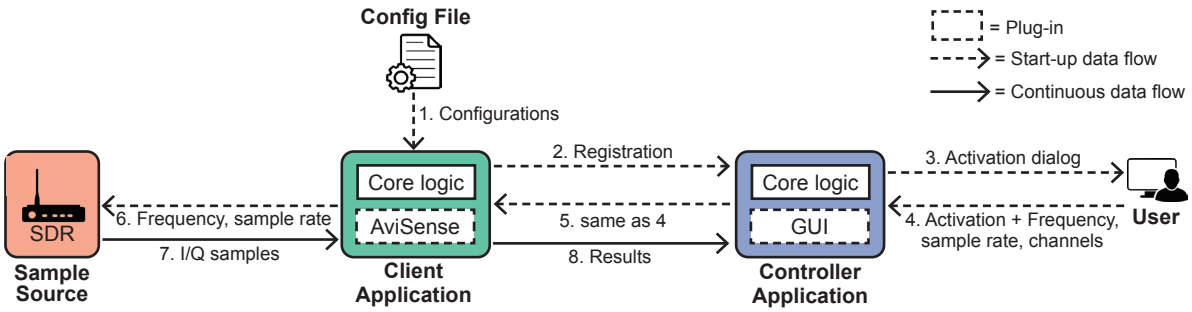


Fig. 23. Our prototype for distributed monitoring with AviSense.

6 AVISENSE PROTOTYPE

We implement AviSense on our general software framework presented in [7]. Our framework provides the core logic to develop a distributed spectrum monitoring system where a controller manages multiple monitoring clients. It only handles the controller-client communication; specific spectrum monitoring capabilities must be

Table 8. Overall classification throughput

Evaluation setup				r_c (%)
Monitored band	**Channels**	n_p	r_o (%)	
VHF-108 (10 MHz)	50 kHz ILS/VOR	4/8/16	100	100
VHF-118 (10 MHz)	25 kHz ACARS	4/8/16	100	100
L-1090 (2 MHz)	N/A	N/A	≤ 25	100
			50	90
			100	50

added by developing corresponding plug-ins. We implement two such client plug-ins for this work; one for our general AviSense architecture (Figure 8) and another for our multi-channel architecture (Figure 14).

6.1 Overview

Figure 23 depicts the components of our end-to-end prototype and their interactions. Upon start-up, our client application loads either the general or multi-channel AviSense plug-in as instructed in a TOML [64] based configuration file. This file also sets different configurations of AviSense modules, e.g., the value for T_o, trained AEs, etc. Next, our client application registers with the controller application. The user can then activate the AviSense plug-in from the controller GUI for a particular channel (or channels if using the multi-channel plug-in).

6.2 Implementation details of AviSense plug-ins

We incorporate a configuration and interfacing (CI) module in both of our plug-ins to handle the interactions with the controller. The main differences between our two plug-ins are in the implementation of the CI module. We implement the rest of the modules such that we can utilize those in both of the plug-ins, with appropriate configuration from the respective CI modules. For our Acquisition module, we implement multiple I/Q sample source options—HackRF One, USRP X310, and USRP B210 using GNU Radio. We also include reading from a 'file' option to incorporate offline testing of AviSense. This option is necessary to evaluate the components of AviSense as well as to determine the r_c on a particular computing device. Interfacing for other SDRs can be integrated into this module. We use Qt's signal-slot mechanism for the communications among modules. We utilize the Vector Optimized Library of Kernels (VOLK) library [66] for efficient implementation of our methods. Our Classification module employs frugally-deep [25] to get predictions from our Keras-based trained AEs.

6.3 Evaluating run-time performance

For the evaluations in this section, we run our client application with AviSense plug-ins on a laptop with Intel Xeon E-2176M (2.7 GHz) processor and the controller application on a separate laptop with Intel Core i7-4810MQ (2.8 GHz) processor. We configure the client and controller applications to communicate over a WiFi network.

6.3.1 Overall classification throughput. r_c reported as part of N_{t_i} is the indicator of the overall classification throughput of AviSense. r_c includes the classification time and the time spent on preprocessing, parameter estimation, sending results to the controller, and system overheads such as thread context-switch and communication among different modules. Besides the computing platform that AviSense is using, r_c also depends on the environment that AviSense is monitoring. If there are fewer signal activities, that is, r_o is low, our computation-heavy

Classification module would not be active frequently. In that scenario, we would get high r_c. Conversely, we can get lower r_c in an environment with high r_o. Therefore, we must evaluate r_c with respect to r_o.

Since it is challenging to accurately control r_o in an over-the-air setup, we use the following custom setup. First, we create several I/Q recordings where each T_o = 1 sec interval has a pre-defined r_o and save those on the ramdisk. Next, we simulate an online environment by running our application with these custom recordings. Using the ramdisk ensures that the process and speed for acquiring I/Q samples are similar to running our application with an SDR.

We present our run-time results in Table 8. For VHF-108 and VHF-118 with our multi-channel plugin, we get r_c = 100% for r_o = 100% with different number of parallel processing paths (n_p). Since the actual signal classification is performed for 25/50 kHz channels, the rate of the I/Q samples that each of our parallel processing paths receive is not very high. Therefore, we can process these without missing any classification. Next, for L-1090 monitoring, we get r_c = 100% for $r_o \leq$ 25%. For higher r_o, our application misses the classification of signal spectrograms. This indicates that our classification cannot keep up with the 2 MHz sample rate for L-1090 when r_o increases beyond 25%. However, L-1090 has bursty transmissions and the highest r_o we encounter for L-1090 is ∼15% in our real-world data collection. Therefore, our prototype would not miss any classification in a regular environment. Our classification throughput can be further improved by using a faster computing device.

6.3.2 SDR interfacing and adaptability. An SDR reports "overflows" and goes into a non-operative state when the processing of the I/Q samples is slower than the sample rate [6, 55]. Encountering such issue at run-time will indicate that AviSense cannot adapt to the available computing resources. Therefore, in this evaluation, we investigate whether an associated SDR experiences such an issue when running our prototype. We simulate a scenario with slow I/Q processing rate as follows. We set our application to monitor a 2 MHz ISM band, to resemble L-1090, band where we continuously transmit a signal to create r_o = 100%. Additionally, we use the AE with the lowest throughput among all of the AEs that we design for this paper to ensure a very slow classification rate. We run our client application in this setup with a HackRF One SDR for about 10 hours. During the course of our experiment, the HackRF One does not report any overflow. This demonstrates that AviSense can adapt to available computing resources without affecting an associated SDR.

7 RELATED WORK

7.1 Aviation spectrum monitoring or anomaly detection

To the best of our knowledge, we are the first to design a versatile spectrum monitoring system for the aviation bands. However, prior studies present anomaly or spoofing attack detection systems for ADS-B [29, 58, 59, 67], which is one of the technologies that uses the L-1090 band. Unlike these studies, we provide a versatile SDR-based system that can be configured to monitor any aviation band and does not require signal demodulation.

7.2 Spectrum anomaly detection

Spectrum anomaly detection has been an important research area. While early works present solutions based on only received power measurements [32, 36, 56], recent ones provide machine learning-based techniques on the time-frequency representation of the observed spectrum [6, 18, 35, 46]. We present an overview of these works and compare with AviSense below.

Feng *et al.* present an AE-based anomaly detection for the FM band [18]. Their technique is evaluated on a small dataset and only for one type of anomaly (i.e., noisy FM signals). In comparison, we evaluate our AEs with a large-scale test data as well as with a wide variety of *anomalous* test signals. Rajendran *et al.* develop an adversarial AE-based spectrum anomaly detection technique [45, 46]. They mainly focus on using a single model to detect anomalies in multiple frequency bands. Such use of a single model is not suitable for AviSense for the following reasons. First, different aviation bands have different bandwidths (e.g., 2 MHz L-1090 band versus

10 MHz VHF-108 band with 50 KHz channels) and comprise different types of signals (e.g., short-lived, bursty signals in L-1090 versus continuous signals in VHF-108) resulting in different challenges. Therefore, using a single model for all of the aviation bands is not practical. Second, one of our design goals for AviSense is to make it easily extensible to other aviation signals and bands, which would not be the case if we use the same model for every band. We would then require retraining for already supported signals and bands along with the new ones. Third, and very importantly, as we demonstrate in Section 5.1, our AEs are 14-20x faster than that of [46], which is necessary to maintain our real-timeliness. Li *et al.* present both Long short-term memory (LSTM) and AE-based anomaly detection methods for the LTE bands [35]. Like ours, they provide an evaluation with different types of anomalies, including the absence of signals, anomalous power levels, and presence of *anomalous* signals. However, they only provide LSTM or AE-based *anomaly* decisions for all of these anomalies without providing means to distinguish them. On the contrary, as we demonstrate in Section 5.4, AviSense facilities distinguishing different types of anomalies. Lastly, our previous study on ISM band signals also includes an AE-based unknown signal detection mechanism [6]. While our past work only focuses on signal classification of the ISM band signals, we present a comprehensive, aviation-focused system in this paper.

In general, we differ from all of these prior ML-based spectrum anomaly detection studies in the following significant way. Prior studies consider ML in isolation and only provide binary decision (anomaly or not) for all types of anomalies, such as, hardware malfunction, unauthorized spectrum use, etc. In contrast, we present a complete, real-time system that extracts actionable spectrum information to detect anomalies and facilitate root cause analysis. Furthermore, we provide a more exhaustive evaluation for *anomalous* signal scenarios.

7.3 Spectrum characterization

Another relevant research area is spectrum characterization where the spectrum utilization with respect to frequency and time is analyzed in real-time [57], via batch processing [70, 72], or from streaming measurements [71]. In contrast to all of these works, we extract more granular spectrum information and identify anomalies in real-time.

7.4 Signal/modulation classification

Wireless signal or modulation classification has been researched heavily in past decades [14, 17, 27, 34, 38, 41, 42, 53, 61, 68]. However, these works are not applicable to AviSense since they mainly deal with supervised multi-class classification. Instead, for AviSense, we need an unsupervised method to distinguish between *normal* and *anomalous* signals.

8 CONCLUSION

We presented AviSense, an SDR-based, *real-time* system for monitoring the aviation frequency bands. AviSense incorporates a novel combination of signal detection, autoencoder-based *anomalous* signal detection, and spectrum characterization to detect and distinguish anomalies. We designed and evaluated AviSense for a diverse set of critical aviation technologies. Furthermore, we designed AviSense as a *versatile* system that can be configured and extended to detect and distinguish anomalies in other aviation bands not explicitly considered in this paper.

ACKNOWLEDGMENTS

We would like to thank Samuel Ramirez for his invaluable insight and guidance on aviation technologies. The support and resources from the Center for High Performance Computing at the University of Utah are gratefully acknowledged. This research made use of the resources of the High Performance Computing Center at Idaho National Laboratory, which is supported by the Office of Nuclear Energy of the U.S. Department of Energy and the Nuclear Science User Facilities under Contract No. DE-AC07-05ID14517.

REFERENCES

[1] Martín Abadi et al. 2015. TensorFlow. https://www.tensorflow.org/.

[2] Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. 2019. Latent space autoregression for novelty detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 481–490.

[3] Aeronautical Radio Inc. (ARINC). 2016. 618-8 Air/Ground Character-Oriented Protocol Specification.

[4] AirNav. 2020. John F Kennedy International Airport. http://www.airnav.com/airport/jfk.

[5] Caglar Aytekin, Xingyang Ni, Francesco Cricri, and Emre Aksu. 2018. Clustering and unsupervised anomaly detection with l 2 normalized deep auto-encoder representations. In *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–6.

[6] Aniqua Baset, Christopher Becker, Kurt Derr, Samuel Ramirez, Sneha Kasera, and Aditya Bhaskara. 2019. Towards Wireless Environment Cognizance Through Incremental Learning. In *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 256–264.

[7] Christopher Becker, Kurt Derr, Samuel Ramirez, Aniqua Baset, and Sneha Kasera. 2019. Plug and Play Flexible Signal Classification and Processing System. In *2019 Resilience Week (RWS)*, Vol. 1. IEEE, 178–184.

[8] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 93–104.

[9] Raghavendra Chalapathy and Sanjay Chawla. 2019. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407* (2019).

[10] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. 2017. Robust, deep and inductive anomaly detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 36–51.

[11] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)* 41, 3 (2009), 1–58.

[12] Jinghui Chen, Saket Sathe, Charu Aggarwal, and Deepak Turaga. 2017. Outlier detection with autoencoder ensembles. In *Proceedings of the 2017 SIAM international conference on data mining*. SIAM, 90–98.

[13] François Chollet et al. 2015. Keras. https://keras.io.

[14] Octavia A Dobre, Yeheskel Bar-Ness, and Wei Su. 2003. Higher-order cyclic cumulants for high order modulation classification. In *MILCOM*, Vol. 1. 112–117.

[15] Pan Du, Warren A Kibbe, and Simon M Lin. 2006. Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. *Bioinformatics* 22, 17 (2006), 2059–2065.

[16] Ettus Research. 2017. The Universal Software Radio Peripheral. https://www.ettus.com/product.

[17] A Fehske, J Gaeddert, and Jeffrey H Reed. 2005. A new approach to signal classification using spectral correlation and neural networks. In *IEEE DySPAN*.

[18] Qingsong Feng et al. 2017. Anomaly detection of spectrum in wireless communication via deep auto-encoders. *The Journal of Supercomputing* (2017).

[19] Great Scott Gadgets. 2018. HackRF one. https://greatscottgadgets.com/hackrf/

[20] GNU Radio 2017. GNU Radio. http://gnuradio.org/.

[21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[22] gr-correctiq [n.d.]. gr-correctiq. https://github.com/ghostop14/gr-correctiq.

[23] Karl Harmer, Gareth Howells, Weiguo Sheng, Michael Fairhurst, and Farzin Deravi. 2008. A peak-trough detection algorithm based on momentum. In *2008 Congress on Image and Signal Processing*, Vol. 4. IEEE, 454–458.

[24] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

[25] Tobias Hermann. 2020. frugally-deep. https://github.com/Dobiasd/frugally-deep.

[26] Heiko Hoffmann. 2007. Kernel PCA for novelty detection. *Pattern recognition* 40, 3 (2007), 863–874.

[27] Steven Siying Hong and Sachin Rajsekhar Katti. 2011. DOF: a local wireless information plane. In *ACM SIGCOMM*.

[28] International Civil Aviation Organization (ICAO). 2006. Annex 10 - Aeronautical Telecommunications - Volume I - Radio Navigational Aids.

[29] Kai Jansen, Liang Niu, Nian Xue, Ivan Martinovic, and Christina Pöpper. 2021. Trust the Crowd: Wireless Witnessing to Detect Attacks on ADS-B-Based Air-Traffic Surveillance. (2021).

[30] James D Johnston. 1988. Transform coding of audio signals using perceptual noise criteria. *IEEE J-STSP* 6, 2 (1988), 314–323.

[31] Ian Jolliffe. 2005. Principal component analysis. *Encyclopedia of statistics in behavioral science* (2005).

[32] Praveen Kaligineedi, Majid Khabbazian, and Vijay K Bhargava. 2010. Malicious user detection in a cognitive radio cooperative sensing system. *IEEE Transactions on Wireless Communications* 9, 8 (2010), 2488–2497.

[33] Hans-Peter Kriegel, Matthias Schubert, and Arthur Zimek. 2008. Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 444–452.

[34] Kaushik Lakshminarayanan, Samir Sapra, Srinivasan Seshan, and Peter Steenkiste. 2009. RFDump: An Architecture for Monitoring the Wireless Ether. In *CoNEXT*.
[35] Zhijing Li, Zhujun Xiao, Bolun Wang, Ben Y Zhao, and Haitao Zheng. 2019. Scaling deep learning models for spectrum anomaly detection. In *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 291–300.
[36] Song Liu et al. 2009. ALDO: An anomaly detection framework for dynamic spectrum access networks. In *IEEE INFOCOM*.
[37] Donald McCallie, Jonathan Butts, and Robert Mills. 2011. Security analysis of the ADS-B implementation in the next generation air transportation system. *International Journal of Critical Infrastructure Protection* 4, 2 (2011), 78–87.
[38] Gihan J Mendis, Jin Wei, and Arjuna Madanayake. 2016. Deep learning-based automated modulation classification for cognitive radio. In *IEEE International Conference on Communication Systems (ICCS)*. IEEE, 1–6.
[39] International Civil Aviation Organization. 2007. International Standards and Recommended Practices, Annex 10: Aeronautical Telecommunications: Surveillance and Collision Avoidance Systems, Vol. 4.
[40] Timothy James O'Shea et al. 2018. Over-the-air deep learning based radio signal classification. *IEEE J-STSP* 12, 1 (2018), 168–179.
[41] Timothy J O'shea, T Charles Clancy, and Hani J Ebeid. 2007. Practical signal detection and classification in gnu radio. In *SDR Forum Technical Conference (SDR)*.
[42] Timothy J O'Shea, Johnathan Corgan, and T Charles Clancy. 2016. Convolutional radio modulation recognition networks. In *International Conference on Engineering Applications of Neural Networks*. Springer, 213–226.
[43] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. 2021. Deep learning for anomaly detection: A review. *ACM Computing Surveys (CSUR)* 54, 2 (2021), 1–38.
[44] Damian Pfammatter, Domenico Giustiniano, and Vincent Lenders. 2015. A software-defined sensor architecture for large-scale wideband spectrum monitoring. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*. 71–82.
[45] Sreeraj Rajendran et al. 2018. SAIFE: Unsupervised Wireless Spectrum Anomaly Detection with Interpretable Features. In *IEEE DySPAN*.
[46] Sreeraj Rajendran, Vincent Lenders, Wannes Meert, and Sofie Pollin. 2019. Crowdsourced wireless spectrum anomaly detection. *IEEE Transactions on Cognitive Communications and Networking* 6, 2 (2019), 694–703.
[47] Sreeraj Rajendran, Wannes Meert, Vincent Lenders, and Sofie Pollin. 2019. Unsupervised wireless spectrum anomaly detection with interpretable features. *IEEE Transactions on Cognitive Communications and Networking* 5, 3 (2019), 637–647.
[48] Gunnar Ratsch, Sebastian Mika, Bernhard Scholkopf, and K-R Muller. 2002. Constructing boosting algorithms from SVMs: An application to one-class classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 9 (2002), 1184–1199.
[49] Shravan Rayanchu, Ashish Patro, and Suman Banerjee. [n.d.]. Airshark: detecting non-WiFi RF devices using commodity WiFi hardware. In *SIGCOMM 2011*. 137–154.
[50] RTCA. 2011. *Minimum Operational Performance Standards for 1090 MHz Extended Squitter: Automatic Dependent Surveillance-Broadcast (ADS-B) and Traffic Information Services-Broadcast (TIS-B)*. Technical Report DO-260B.
[51] Mayu Sakurada and Takehisa Yairi. 2014. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*. 4–11.
[52] Salvatore Sanfilippo and M Robb. 2014. dump1090.
[53] Shamik Sarkar, Milind Buddhikot, Aniqua Baset, and Sneha Kumar Kasera. 2021. DeepRadar: a deep-learning-based environmental sensing capability sensor design for CBRS. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. 56–68.
[54] Harshad Sathaye, Domien Schepers, Aanjhan Ranganathan, and Guevara Noubir. 2019. Wireless attacks on aircraft instrument landing systems. In *28th USENIX Security Symposium*. 357–372.
[55] Erick Schmidt, David Akopian, and Daniel J Pack. 2017. Development of a Real-Time Software-Defined Radio GPS Receiver Exploiting a LabVIEW-based Instrumentation Environment. *IEEE Transactions on Instrumentation and Measurement* (2017).
[56] Anmol Sheth, Christian Doerr, Dirk Grunwald, Richard Han, and Douglas Sicker. 2006. MOJO: A distributed physical layer anomaly detection system for 802.11 WLANs. In *Proceedings of the 4th international conference on Mobile systems, applications and services*. 191–204.
[57] Lixin Shi, Paramvir Bahl, and Dina Katabi. 2015. Beyond sensing: Multi-ghz realtime spectrum analytics. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*. 159–172.
[58] Martin Strohmeier, Vincent Lenders, and Ivan Martinovic. 2015. Intrusion detection for airborne communication using PHY-layer information. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 67–77.
[59] Martin Strohmeier, Ivan Martinovic, Markus Fuchs, Matthias Schäfer, and Vincent Lenders. 2015. Opensky: A swiss army knife for air traffic security research. In *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*. IEEE, 4A1–1.
[60] Martin Strohmeier, Matthias Schäfer, Rui Pinheiro, Vincent Lenders, and Ivan Martinovic. 2016. On perception and reality in wireless air traffic communication security. *IEEE transactions on intelligent transportation systems* 18, 6 (2016), 1338–1357.
[61] Ananthram Swami and Brian M Sadler. 2000. Hierarchical digital modulation classification using cumulants. *IEEE Transactions on communications* 48, 3 (2000), 416–429.

[62] Jian Tang, Zhixiang Chen, Ada Wai-Chee Fu, and David W Cheung. 2002. Enhancing effectiveness of outlier detections for low density patterns. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 535–548.
[63] David MJ Tax and Robert PW Duin. 2004. Support vector data description. *Machine learning* 54, 1 (2004), 45–66.
[64] TOML [n.d.]. TOML. https://toml.io/en/.
[65] Hiroshi Tsurumi and Yasuo Suzuki. 1999. Broadband RF stage architecture for software-defined radio in handheld terminal applications. *IEEE Communications Magazine* 37, 2 (1999), 90–95.
[66] VOLK 2017. Vector Optimized Library of Kernels. http://libvolk.org/.
[67] Xuhang Ying, Joanna Mazer, Giuseppe Bernieri, Mauro Conti, Linda Bushnell, and Radha Poovendran. 2019. Detecting ADS-B spoofing attacks using deep neural networks. In *2019 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 187–195.
[68] Tevfik Yucek and Huseyin Arslan. 2006. Spectrum characterization for opportunistic cognitive radio systems. In *IEEE MILCOM*.
[69] Tevfik Yucek and Huseyin Arslan. 2009. A survey of spectrum sensing algorithms for cognitive radio applications. *IEEE Comm. surveys & tutorials* (2009).
[70] Yijing Zeng, Varun Chandrasekaran, Suman Banerjee, and Domenico Giustiniano. 2019. A framework for analyzing spectrum characteristics in large spatio-temporal scales. In *The 25th Annual International Conference on Mobile Computing and Networking*. 1–16.
[71] Mariya Zheleva, Petko Bogdanov, Timothy Larock, and Paul Schmitt. 2018. AirVIEW: Unsupervised transmitter detection for next generation spectrum sensing. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 1673–1681.
[72] Mariya Zheleva, Ranveer Chandra, Aakanksha Chowdhery, Ashish Kapoor, and Paul Garnett. 2015. Txminer: Identifying transmitters in real-world spectrum measurements. In *2015 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE, 94–105.
[73] Chong Zhou and Randy C Paffenroth. 2017. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 665–674.
[74] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. 2018. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*.