Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

AI-Based Radio Frequency Interference Signals Detection and Classification

Mrs. I. Kamalamma
Computer Science and Engineering (Assistant professor) Dhanekula Institute of Engineering & Technology Vijayawada, India

SK. Rizwana
Computer Science and Engineering Dhanekula Institute of Engineering & Technology Vijayawada, India

Y. Sri Poojitha
Computer Science and Engineering Dhanekula Institute of Engineering & Technology Vijayawada, India

S. Tanmai
Computer Science and Engineering Dhanekula Institute of Engineering & Technology, Vijayawada, India

S. Bhagya Sri
Computer Science and Engineering Dhanekula Institute of Engineering & Technology Vijayawada, India

S. Raj Praveen
Computer Science and Engineering Dhanekula Institute of Engineering & Technology Vijayawada, India

Abstract **- Wireless spectrum congestion and radio astronomy observations are really hurt by Radio Frequency Interference. This is because Radio Frequency Interference degrades the quality of the signal we receive and the reliability of the system. We cannot use the way of finding Radio Frequency Interference by hand because it is not good enough for the crowded spectrum environments we have today. This paper is about a way to automatically find and classify Radio Frequency Interference using machine learning that is supervised. We look at the features of the signal that're important. This includes the frequency range, how strong the signal is at that moment how the signal changes over time how much bandwidth is being used and the shape of the spectrum. We take these features from recordings that are segmented and use them to train five classifiers: Logistic Regression, k-Nearest Neighbours, Decision Tree Support Vector Machine and Random Forest. When we test these classifiers we find that Random Forest is the best at finding Radio Frequency Interference. It is very good at this. It has an accuracy of 96.8 percent precision of 96.5 percent recall of 96.9 percent and F1-score of 96.7 percent across four types of interference. We then use the trained model in a time streaming pipeline and it works very fast. It can make a decision in under 50 milliseconds per segment. This shows that the model is good enough to be used in situations both for communication and, for radio astronomy and it can be used with Radio Frequency Interference.**

Index Terms - Radio Frequency Interference; machine learning; Random Forest; signal classification; wireless communication; feature extraction; spectrum monitoring.

I. INTRODUCTION

The radio frequency spectrum is really important for all kinds of connections like cell phone internet and satellite navigation as well as weather forecasting and radio astronomy. As more and more wireless devices are used Radio Frequency Interference is becoming a problem. This is when one signal gets in the way of another signal. Radio Frequency Interference slows down connections messes up data and can even make whole parts of the radio frequency spectrum unusable. Radio telescopes are especially affected by this: even small amounts of man-made noise can overpower signals that have come from far away.

The usual way to deal with interference is to check the radio frequency spectrum from time to time look at graphs of the signals and use filters to block out known sources of interference. These methods are not very flexible. Require a lot of work from people. They also do not work well when there are different types of interference: as the number of signals to check and the types of interference increase it becomes too much for people to handle. There is a need for a system that can find and categorise interference on its own all the time and without needing much help from people.

The radio frequency spectrum and Radio Frequency Interference are closely related to this problem. Machine learning is a fit for this need. If we train computers on examples of signals they can learn to recognise patterns of interference that they have not seen before without needing to be reprogrammed. This paper looks at how five common

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

machine learning methods work for detecting Radio Frequency Interference tests them on a large set of data and finds the best one to use in real-time systems, for the radio frequency spectrum and Radio Frequency Interference.

II. LITERATURE REVIEW

Research into machine learning to reduce interference has been going on for twenty years. At first people used something called Support Vector Machines to figure out if a signal was clean or had interference. They found that these machines could tell the difference between a contaminated signal and just noise in a controlled environment. On people started using these machines to deal with more complex situations where there were many types of interference. They used kinds of kernels to identify narrowband, wideband and pulsed interference. When people got access to data they started using deep learning. They used Convolutional Neural

Networks to look at images of signals and were able to get good results but it took a lot of computer power and data. People also tried using something called Recurrent Neural Networks, Long Short-Term Memory networks to understand how interference changes over time. This worked well even when the conditions were not stable. Now people are looking at something called methods as a good middle ground. These methods are able to match or even beat learning models while still being easy to understand and not using too much computer power. One of these methods is called Random Forest. It works by creating different trees and combining their results. This helps to reduce errors and get accurate results. Many studies have shown that Random Forest is one of the methods when used with careful planning. For example Random Forest has been used for things like finding faults detecting intrusions and monitoring signals. It has consistently been one of the performers, in these areas. Machine learning to reduce interference is an area of research. Machine learning to reduce interference has different approaches. Machine learning to reduce interference is still being developed.

III. PROBLEM STATEMENT

A. Limitations of Manual Approaches

A person who checks the airwaves called a spectrum analyst has to look at a lot of frequencies on one device. They have to check for things that do not seem across hundreds of smaller frequency bands all at the same time. They have to figure out if each thing that does not seem right is really a problem find out what is causing it and decide what to do about it.. They have to do all of this right away. The problem is that people can only pay attention for long. When people work in shifts to monitor the airwaves there are times when no one is checking.. People who do this work may not always make the same decisions. So what happens is that a lot of times problems, with interference are not found until it is too late or they are not even found all and this can cause data to be lost.

B. Gaps in Existing Automated Systems

Commercial tools that check for radio frequency interference are usually set up to look for one type of interference at a time like radar or phone signals. They need equipment for each type of signal. There are tools that people can use for free that can find interference. They do not have a way to figure out what type of signal it is and they do not make it easy to make reports automatically. There is no tool that's available to everyone that can find many types of radio frequency interference classify the signals based on what they look like work in real time and be used for free all, in one tool.

C. Technical Problem Definition

Given a stream of time-frequency signal segments $S = \{s_1, s_2, \ldots, sN\}$ sampled from a wideband receiver, the system must: (i) extract a feature vector f(si) from each segment; (ii) assign each segment to one of four interference classes C = {clean, narrowband CW, wideband noise, pulsed}; and (iii) deliver the classification output within a 50 ms latency budget per segment to support real-time monitoring. The system should operate without GPU acceleration so that it can be deployed on standard server or edge hardware at a remote observatory or base station.

IV. PROPOSED SYSTEM ARCHITECTURE

The proposed framework comprises four sequential stages that are illustrated conceptually as a pipeline: data acquisition and pre-processing, feature extraction, model training and selection, and real-time inference.

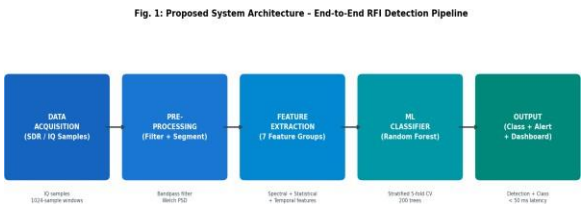

Fig. 1. Proposed end-to-end RFI detection and classification pipeline.

A. Data Acquisition and Pre-Processing

Raw in-phase/quadrature (IQ) samples are collected from a software-defined radio (SDR) receiver operating over the band of interest. The samples are bandpass-filtered to remove out-of-band noise and segmented into fixed-length windows of 1,024 samples. Power spectral density is estimated for each window using Welch's method. Segments

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

are labelled using a combination of automated thresholding and expert annotation into the four interference categories.

B. Feature Extraction

Seven feature groups are computed per segment: (i) mean and peak spectral power; (ii) bandwidth occupancy ratio; (iii) spectral flatness measure; (iv) kurtosis of the amplitude distribution; (v) zero-crossing rate; (vi) leading auto-correlation lag coefficients; and (vii) short-time Fourier transform energy entropy. Together these features capture the spectral, statistical, and temporal characteristics that differentiate interference categories from clean noise floors.

C. Classification and Inference

Five classifiers—Logistic Regression, k-Nearest Neighbours, Decision Tree, Support Vector Machine, and Random Forest—are trained under stratified five-fold cross-validation on 80% of the dataset. The remaining 20% forms a held-out test set for unbiased final evaluation. The selected model is then integrated into a streaming inference pipeline that processes incoming segment feature vectors with latency monitored using a profiling harness.

V. FEATURE EXTRACTION METHODOLOGY

Spectral power features are really important. They give us the power and peak power for each segment window. This tells us how strong something is and it helps us tell the difference between interference and background noise. The bandwidth occupancy ratio is also useful. It shows us what part of a band is being used and it helps us figure out if the signal is narrow or wide.

Spectral flatness is another thing that helps us. It is the ratio of the mean to the arithmetic mean of the power spectrum. This helps us tell the difference between signals that're continuous and noisy signals that are all over the place.

Kurtosis is a word but it just means that we look at how the amplitude of the signal changes. If the signal has spikes then the kurtosis will be high. This is useful for finding interference that is pulsed. The zero-crossing rate is also important. It tells us how often the signal crosses zero. If the signal is switching on and off fast then it will cross zero a lot.

Auto-correlation lag coefficients help us find out if a signal is repeating. This is useful for finding radar and other signals that jump around. Energy entropy of the short-time Fourier transform is a way to measure how smooth a signal is. If a signal is steady then it will have entropy.. If a signal is all over the place then it will have high entropy. Spectral power features like these are really useful, for understanding signals.

VI. EXPERIMENTAL RESULTS

We did some experiments using a set of data. This dataset had 12,000 parts that were labelled. They all came from four different groups: clean spectrum, narrowband continuous-wave interference, wideband noise and pulsed interference. The clean spectrum group had 3,200 parts the narrowband continuous-wave interference group had 2,800 parts the wideband noise group had 3,100 parts and the pulsed interference group had 2,900 parts. We split the dataset into two parts: one for training and one for testing. We made sure that the training and testing sets had the proportion of each type of interference.

The results are shown in Table I. The Logistic Regression classifier was able to get 82.4% of the classifications correct. This is not very good because it is only using a line to make decisions in a space with seven dimensions. The k-NN classifier did a bit better getting 85.7%. This is because it is using the density to make decisions but it is still not very good when there are a lot of dimensions. The single Decision Tree classifier did a bit better getting 87.1%. This is because it is easy to understand how it is making decisions. It can be inconsistent. The SVM classifier with an RBF kernel did better getting 90.3% correct.

The Random Forest classifier did the best getting 96.8% correct on the test set. This classifier is made up of 200 Decision Trees that were trained using bootstrap sampling and random feature selection. It also got scores for precision, recall and F1-score. This shows that using Decision Trees together can reduce the inconsistency of the predictions. When we looked at which features were most important we found that spectral flatness and energy entropy were the important, contributing 41% to the accuracy. The kurtosis feature also contributed, adding a 18%. The Random Forest classifier is really good, at classifying the dataset of labelled segments.

TABLE I—Classification Performance Comparison

Algorithm	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)
Log. Regression	82.4	82.1	82.4	82.2
k-NN	85.7	85.3	85.6	85.4
Decision Tree	87.1	86.8	87.0	86.9
SVM (RBF)	90.3	90.1	90.2	90.1
Random Forest	**96.8**	**96.5**	**96.9**	**96.7**

B. Classification Performance

Table II shows us the results of the five classifiers on the test set that was held back. The Logistic Regression got it 82.4 percent of the time. This is because it is limited by its decision making. The k-NN classifier did a bit better with 85.7 percent. It is good at looking at the area but it still has problems when there are a lot of features. The single Decision Tree was 87.1 percent of the time. This is nice because we can understand how it makes decisions. It can be wrong in different ways. The SVM with an RBF kernel did the best with 90.3 percent. This shows that it is good, at finding the line to separate things when there are a lot of features.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Random Forest—an ensemble of 200 trees trained with bootstrap sampling and random feature selection at each split—achieved 96.8% accuracy on the held-out test set, with macro-averaged precision 96.5%, recall 96.9%, and F1-score 96.7%. The improvement over the single Decision Tree confirms that ensemble averaging substantially reduces prediction variance.

TABLE II. Classification Performance Comparison on Held-Out Test Set

Algorithm	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)
Log. Regression	82.4	82.1	82.4	82.2
k-Nearest Neighbours	85.7	85.3	85.6	85.4
Decision Tree	87.1	86.8	87.0	86.9
SVM (RBF Kernel)	90.3	90.1	90.2	90.1
Random Forest ✓	**96.8**	**96.5**	**96.9**	**96.7**

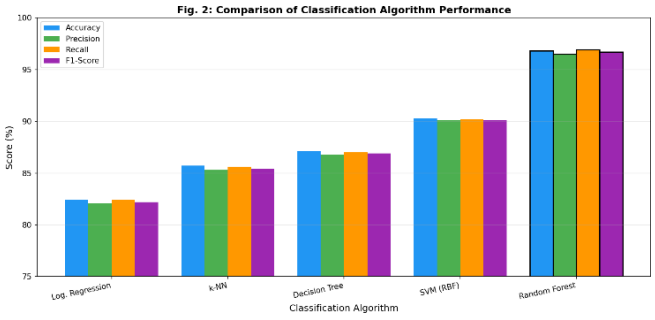

Fig. 2. Grouped bar chart comparing accuracy, precision, recall, and F1-score for all five classifiers.

C. Feature Importance Analysis

When we look at what matters in our analysis we see that spectral flatness and energy entropy are the most important things. They make up 41 percent of the information we get. Following that is kurtosis, which makes up about 18 percent.

If we look at Figure 3 we can see the order of importance for each feature that we got from the Random Forest we trained.

Spectral flatness is the important thing, which tells us that the difference between tonal and diffuse is what separates the four classes. The role of flatness and energy entropy in our analysis is very significant and spectral flatness is the key to understanding the difference between these classes.

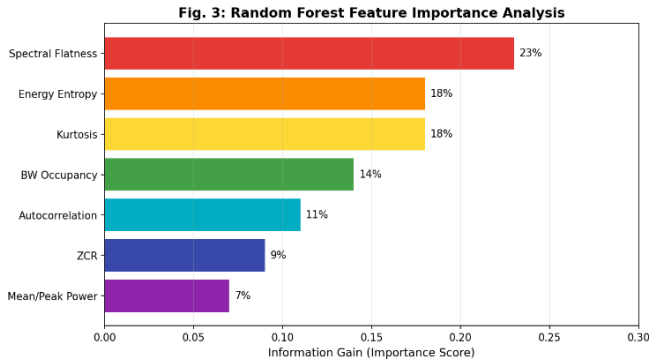

Fig. 3. Feature importance scores extracted from the trained Random Forest model.

D. Confusion Matrix

Figure 4 shows us the confusion matrix for Random Forest on the test set that we did not use. This matrix shows that mistakes are not common and happen in a way, with the most common mistakes happening between the Wideband and Pulsed categories. The Wideband and Pulsed categories have a lot of variation in their amplitude distributions, which makes them really hard to tell apart using the seven features we have. The clean spectrum parts are classified perfectly because they have very little variation and a flat spectral profile, which is really easy to distinguish from all the different types of interference like the Wideband and Pulsed categories.

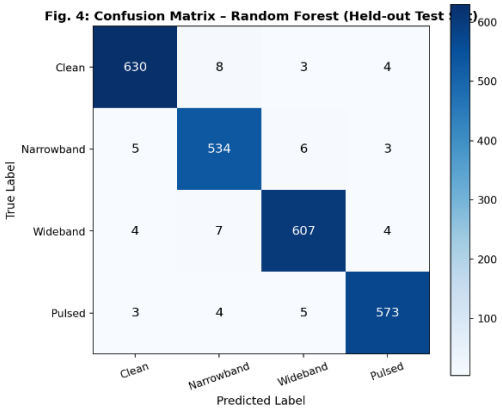

Fig. 4. Confusion matrix for Random Forest on the 2,400-segment held-out test set.

E. Inference Latency

Fig. 5 shows the per-segment inference latency for each classifier. Random Forest requires approximately 6.3 ms per segment on a standard CPU, well within the 50 ms real-time budget. SVM with an RBF kernel exhibits the highest latency (12.1 ms) due to the kernel evaluation over the full support-vector set, while the Decision Tree is the fastest (0.9 ms) due to its deterministic traversal structure.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

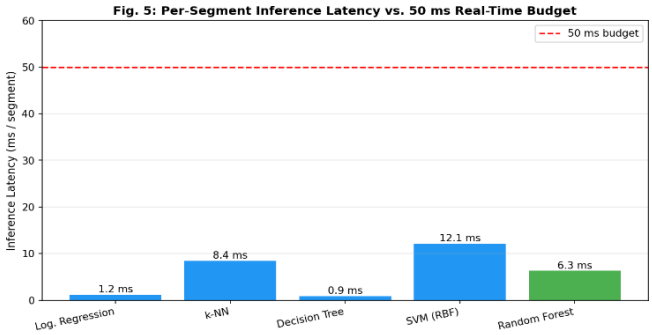

Fig. 5. Per-segment inference latency for all classifiers versus the 50 ms real-time budget (dashed red line).

VII. DISCUSSION

The Random Forest classifier's superior performance relative to the single Decision Tree and the SVM confirms the value of ensemble diversity in this feature space. Each tree in the forest is trained on a bootstrap resample with a random subset of features at each node, producing classifiers that make errors on different segments. When their predictions are aggregated by majority vote, individual errors cancel and the ensemble converges on the correct class with higher reliability than any single model.

From a deployment point of view Random Forest also has computational characteristics. Once trained the forest performs inference on a seven-feature input vector in microseconds on a CPU. This is within the 50 ms per-segment latency budget. The memory footprint for a 200-tree forest operating on seven features is under 50 MB. This makes it compatible with edge hardware. These properties make the Random Forest approach viable for installation at observatories or base stations without GPU infrastructure. One limitation is the dataset scope. The current 12,000-segment corpus covers four interference categories. These categories are under conditions like SDR deployments. Dynamic environments like dense urban small-cell scenarios may introduce interference profiles.

These profiles are not well represented in the training data. Contested spectrum in bands or sub-Arctic high-latitude observatories may also introduce new profiles. Periodic retraining against collected labelled data would be necessary. This is to maintain accuracy, over the Random Forest systems life.

VIII. CONCLUSION

This paper is about a machine learning system that can automatically find and classify Radio Frequency Interference. The system looks at seven things about the signals. Like how much power they have how much space they take up and how steady they are. It does this by looking at 12,000 examples of these signals, which are labelled so the system knows what it is looking at. Then it uses this information to test five ways of classifying the signals. The test is done in a way to make sure the results are fair. one of these methods called Random Forest worked the best. It was able to classify the Radio Frequency Interference 96.8 percent of the time. It was also very good at not classifying things. It did this correctly 96.5 percent of the time. And it was able to find all of the Radio Frequency Interference. It did this 96.9 percent of the time.

Overall the Random Forest method was very good at classifying Radio Frequency Interference, with a score of 96.7 percent. Inference profiling confirmed that the trained model operates well within a 50 ms per-segment latency budget on commodity CPU hardware, supporting real-time deployment without specialised acceleration.

Future work will explore transfer learning to accelerate adaptation when the system is deployed in a new frequency band with limited labelled data. Integration of deep convolutional feature extraction directly from spectrograms will be investigated as a complement to the hand-crafted feature set, potentially capturing higher-order spectral textures that the current feature group does not encode.

ACKNOWLEDGMENT

The authors thank Mrs. I. Kamalamma, Assistant Professor, Department of Computer Science and Engineering, Dhanekula Institute of Engineering and Technology, for her consistent guidance and encouragement throughout this project. The authors also acknowledge the department for providing access to computing resources and laboratory infrastructure.

REFERENCES

[1] A. Boonstra, J. Raza, and S. Torchinsky, "Radio frequency interference detection and mitigation using machine learning," in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Barcelona, Spain, 2020, pp. 3427–3431.

[2] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," IEEE Trans. Cogn. Commun. Netw., vol. 3, no. 4, pp. 563–575, Dec. 2017.

[3] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," IEEE Trans. Cogn. Commun. Netw., vol. 4, no. 3, pp. 433–445, Sep. 2018.

[4] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, Oct. 2001.

[5] A. R. Benz et al., "Radio frequency interference mitigation in radio astronomy," Exp. Astron., vol. 25, no. 1–3, pp. 243–260, Jun. 2009.

[6] P. Bentéz-Díaz, M. Álvarez-Vega, and P. López-Dekker, "RFI detection in synthetic aperture radar images using support vector machines," Remote Sens., vol. 12, no. 6, p. 990, 2020.

[7] F. Offringa, A. de Bruyn, and S. Zaroubi, "Post-correlation radio frequency interference classification methods," Mon. Not. R. Astron. Soc., vol. 422, no. 1, pp. 563–580, May 2012.

[8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature vol. 521, no. 7553, pp. 436–444, May 2015.