

WHITE PAPER

MACHINE LEARNING & RF SPECTRUM INTELLIGENCE GATHERING

Prepared by **Dr Michael Knott**

CRFS

EXTRAORDINARY
RF TECHNOLOGY

This white paper describes how CRFS is solving customer problems more effectively, by using ML to automate pattern recognition processes as part of a suite of EW tools. We start by listing some guiding principles that need to apply, in order for an implementation of ML to be most practically useful for the purpose. We then turn to three tasks in spectrum intelligence gathering that can be approached with ML. For each case, we describe the technique used and present some results.



CONTENTS

1	**Introduction**	**03**
2	**Guiding principles**	**04**
3	**Machine learning for automated signal classification**	**05**
	3.1 A traditional technique: decision trees	05
	3.2 Machine learning applied to the same problem	06
	3.3 Pair plots	07
	3.4 Training: it's all about the database	09
	Test results: it's all about the training	10
	What does it look like in real life?	12
	Why use machine learning for signal classification?	14
4	**Machine learning for signal detection**	**15**
5	**Machine learning for spectrum monitoring**	**17**
6	**Efficiency of ml processing**	**18**
7	**Looking to the future: feature learning**	**18**
8	**Conclusion**	**19**
9	**Why crfs?**	**19**
Legal information		**20**
Disclaimer		**20**

1. INTRODUCTION

Many applications that are central to RF spectrum intelligence gathering require some sort of pattern recognition. For example, to classify a signal by type we need to identify the particular pattern associated with the modulation, while to recognize that there is an interesting signal present in received data, we need to distinguish between pattern and noise.

It is desirable to automate these applications as far as possible, for efficiency and to avoid errors caused by fatigue. However, pattern recognition is a task that human brains are good at, while computers, in their traditional role of applying predetermined algorithms, are not. This has made automation of the process challenging. Machine learning (ML), which is a mature set of mathematical techniques that allow computers to learn from data using constructs such as artificial neural networks, is better suited to the problem.

The past ten years have seen a large amount of investment in ML from companies such as Google and Baidu. One estimate is that between $26 billion and $39 billion was invested in artificial intelligence in 2016, of which the bulk involved ML[1]. This has been stimulated by the realization that ML can help solve challenging problems in areas such as search optimization, advertisement placement, face recognition and driverless car technology. New tool sets for ML (for example, Google's TensorFlow) have been developed and made publicly available under open source licensing policies. Research using these tool sets has advanced the fields of machine learning and neural networks.

This white paper describes how CRFS is solving customer problems more effectively, by using ML to automate pattern recognition processes as part of a suite of EW tools. We start by listing some guiding principles that need to apply, in order for an implementation of ML to be most practically useful for the purpose. We then turn to three tasks in spectrum intelligence gathering that can be approached with ML. For each case, we describe the technique used and present some results.

The tasks are: (1) signal classification, for which we use a simple feedforward neural network; (2) signal detection, using a convolutional neural network to look for patterns in the data; (3) spectrum monitoring, using convolutional and unsupervised neural networks for anomaly detection.

Artificial neural networks (ANNs) are networks of virtual neurons, inspired by the structure of the human brain. The output of each neuron depends on the weighted inputs it receives from other neurons. The arrangement of the network is fixed, but the behavior of the neurons is governed by numbers, which are modified as the network learns. This paper is not intended to be an introduction to ANNs or to machine learning in general, but some details are mentioned in passing.

We also discuss how the available processing power affects what can be achieved by ML in spectrum intelligence gathering. We finish with a look at the future, and at some of the ways in which this application of ML could be developed further, to make spectrum intelligence gathering more reliable and more truly automated.

1 *Artificial Intelligence: The Next Digital Frontier? (McKinsey Global Institute, 2017)*

2. GUIDING PRINCIPLES

In order to produce a deployable ML solution for signal classification and detection, we follow some guiding principles.

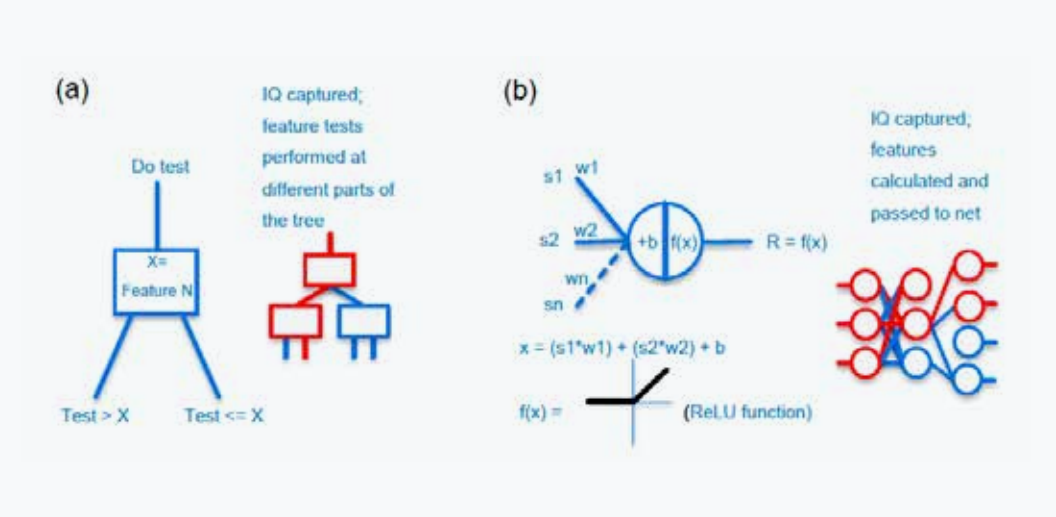
- Applied neural networks should be able to accept either live or archived data. This allows the solution to be used to reduce the number of false alarms or missed events by a human operator looking at live systems in operation or, alternatively, to search through terabytes of archived data.
- Computational requirements should be modest, so that the solution can run on embedded distributed sensor and transmitter platforms: analyzing data at source reduces data connectivity requirements and enables more rapid notification of alarms.
- It should be possible to update a trained neural network by training it further using a new database of signals.

3. MACHINE LEARNING FOR AUTOMATED SIGNAL CLASSIFICATION

Automated signal classification is the first problem to which we will apply machine learning in this paper. The system is given an unknown signal, and the requirement is to determine its modulation type and protocol. For example, given a TETRA signal it should be able to work out that the modulation is π/4 DQPSK and the protocol is TETRA.

3.1 A TRADITIONAL TECHNIQUE: DECISION TREES

One traditional means of applying algorithmic methods to automated signal classification is the decision tree (DT). Here, multiple numerical features are extracted from a captured signal. Features can range from the simple (such as an average) to the much more complicated (such as the results of a Fourier analysis). Decisions are made consecutively using each of these features, in a predetermined order that depends on earlier decisions, resulting in a branching 'tree' structure (Error! Reference source not found. (a)). Thresholds for the decisions are varied manually to optimize the decision process, using a database of signals of known type. The DT decision process is fundamentally linear, so the technique will struggle with some feature sets.



<< ***Figure 1***
Nodes for feature-based signal classification: (a) Node of decision tree method; (b) Node of artificial neural net, showing weighted inputs and ReLU activation function

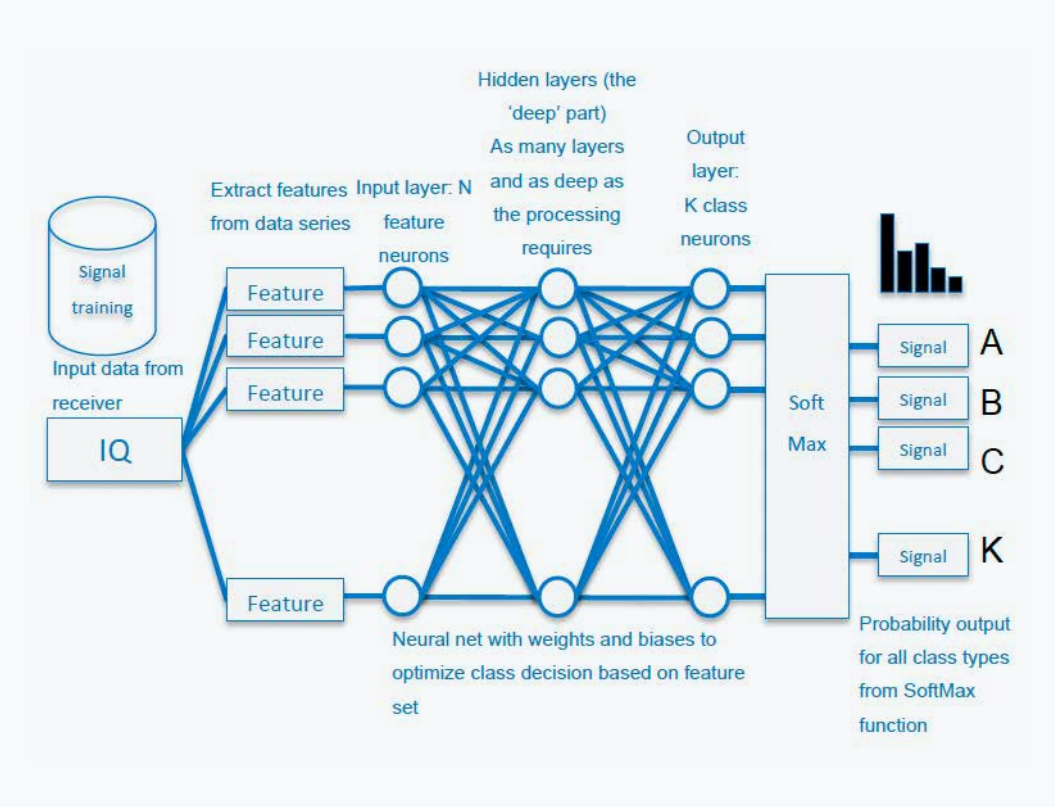
3.2 MACHINE LEARNING APPLIED TO THE SAME PROBLEM

Machine learning (ML) can also use features extracted from the captured signal. We will show the results of applying features to an artificial neural network (ANN) in order to classify signals. We use a feedforward neural network, which is a network in which the outputs from each layer of neurons are connected only to the inputs of succeeding layers, with no feedback during the decision process. Features are applied to the input layer, there is at least one hidden layer, and an output layer feeds into a voting function (SoftMax) which estimates the probability of each signal type (Figure 2).

If one signal type is much more probable than all the others, then the result is definite and easy to interpret. If a number of different signal types are nearly equally probable, the ANN will correctly return an uncertain answer, instead of imparting a false certainty. This is in contrast with DT, which forces a definite answer regardless of how ambiguous the signal might be.

Individual neurons produce an output that is calculated by feeding the weighted sum of their inputs into a nonlinear rectified linear unit (ReLU) activation function (Figure 1(b)); this nonlinearity can allow the ANN to perform well with feature sets that DT struggles with. We train the network on a database of signals of known type. Training follows the backpropagation method, in which the weight of a connection is modified, at each training step, by an amount proportional to the gradient of the output error with respect to the weight. Gradients are calculated first for the output layer and then propagated back to the hidden layers.

The features that are used by this application of ML are chosen by engineers, just as with DT, rather than being discovered by the ML itself. Features are selected to be effective at distinguishing between different signal types.



<< ***Figure 2*** *Feedforward neural network for feature-based signal classification*

3.3 PAIR PLOTS

To illuminate some of the decision making in both DT and ML, we can plot individual signals according to the values that they exhibit of pairs of features; the pair plots let us see how different types of signals can form clusters and how they can be distinguished from one another using different combinations of features.

Figure 3 shows some idealized pair plots. These are illustrative and were not derived from real data. The dividing lines between signal types are also idealized and simplified, but they serve to illustrate the difference between linear and nonlinear fitting.

In Figure 3(b), the two features are very good for separating the signal types; in this case DT, with a linear decision fit, would perform as well as ML. Plot (c) shows a situation where the two features are unsuitable for distinguishing the signal types because they contain no information that is useful for the classification process. Plot (a) shows a case where ML would outperform DT because it allows nonlinear decision fitting to the data.



<< ***Figure 3*** *Idealized pair plots*

A problem that can occur with ML (and also with DT) is overfitting. This is when the ML learns the characteristics of a specific set of training data, rather than the characteristics of the type that we intend it to learn, so that it does not successfully classify other data sets. In a pair plot such as Figure 3(a), this might manifest itself as a convoluted dividing line that encompasses all the individual points of signal type 1, but is not a reasonable outline for the cluster. We avoid overfitting by using pre-developed tools from TensorFlow.

Figure 4 shows some real examples of pair plots for a larger range of features and signals. For the classification of 10 signal types, 19 features are extracted from the data set. The pair plots are much more complicated than the idealized examples of Figure 3. ML fits all of these automatically, whereas the DT method would require manual fitting.

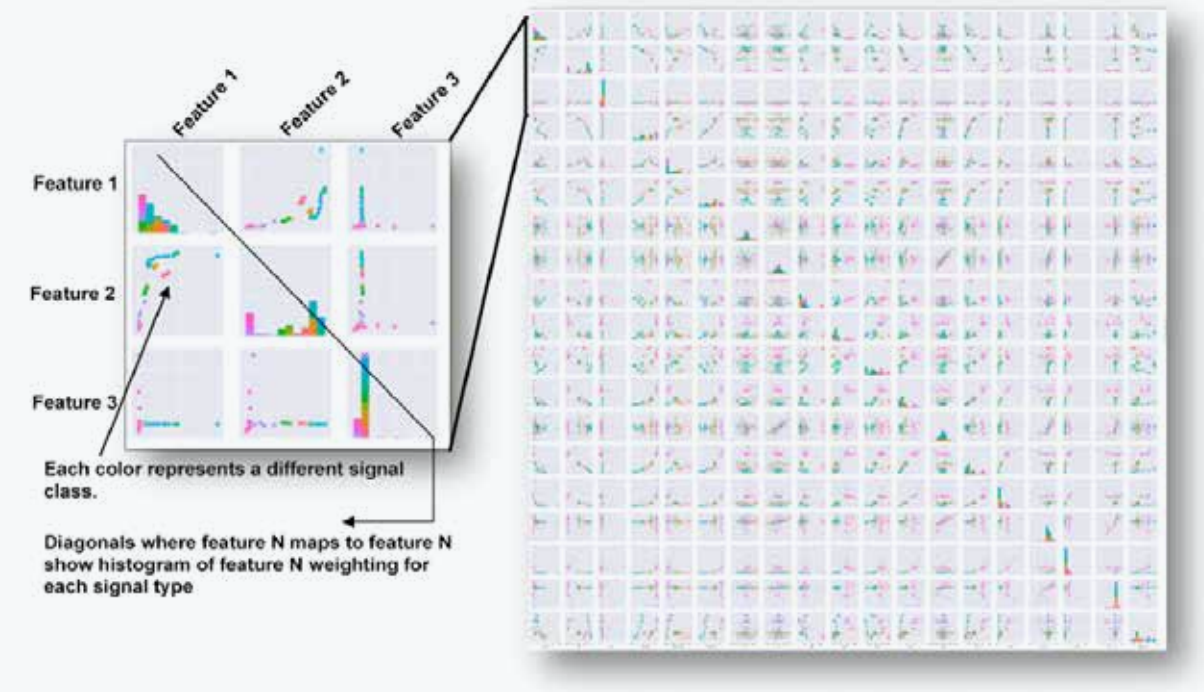


Figure 4 *Real pair plots*

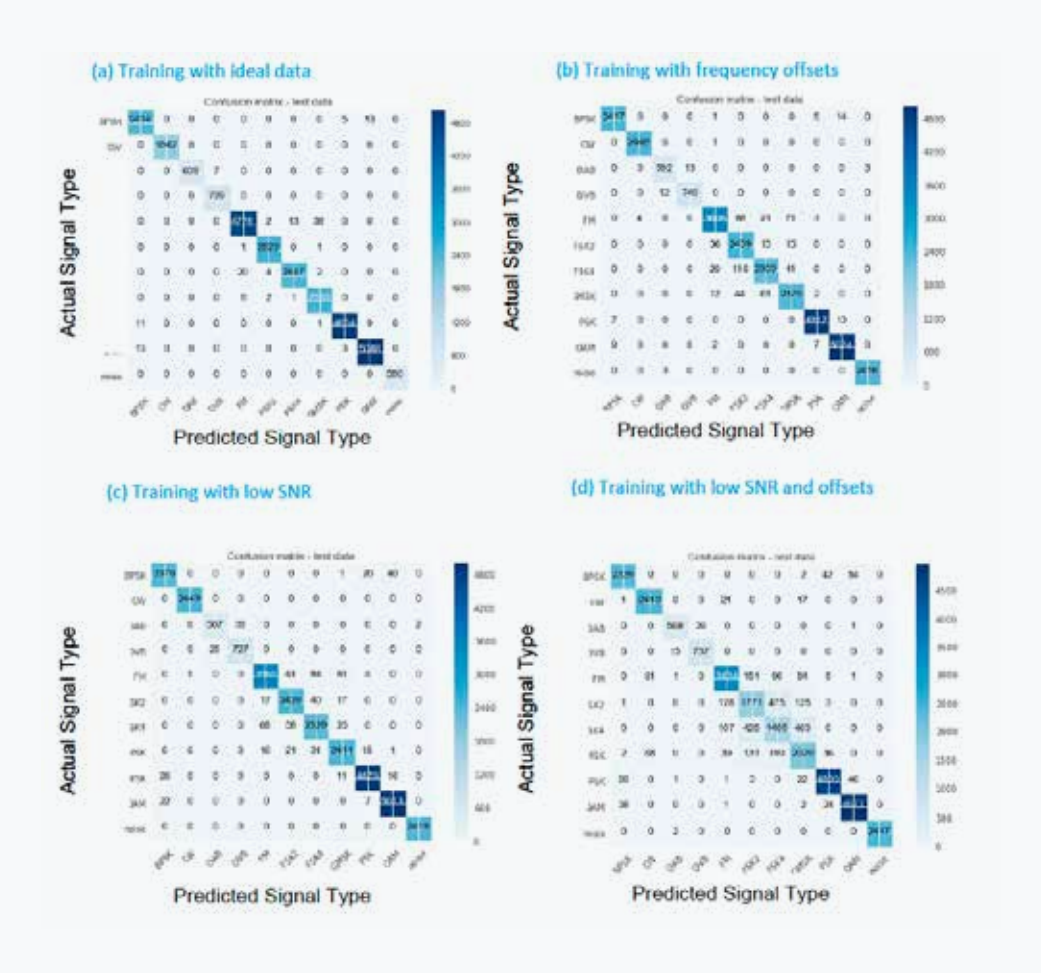
3.4 TRAINING: IT'S ALL ABOUT THE DATABASE

The signal database is crucial to the performance of ML (as it is also for DT). The decision process must be optimized for the signal types in general, not for the signal types under some limited set of conditions. This means that the database should contain signals received under a range of different conditions, representative of the conditions under which the trained network is expected to function.

To protect against unwittingly training a network that is unsatisfactory because it is poorly fitted, or overfitted, to the training data, the data is divided into a training set and a test set. The ANN is trained using only the training set, then tested using the test set to make sure it is effective with that data too.

3.5 TEST RESULTS: IT'S ALL ABOUT THE TRAINING

A useful way to evaluate the performance of an ANN is to enter the test results into a confusion matrix (Figure 5). This is a matrix showing the frequency with which a signal of each type is identified as belonging to each of the possible types. The more of the identifications that lie on the diagonal of the matrix, the better the performance of the network. Confusion matrices allow us to see how reliable the classifier is and, by showing the mistakes in detail, they also help us to decide what new features might make it more reliable.



<< ***Figure 5***
Confusion matrices for different data sets

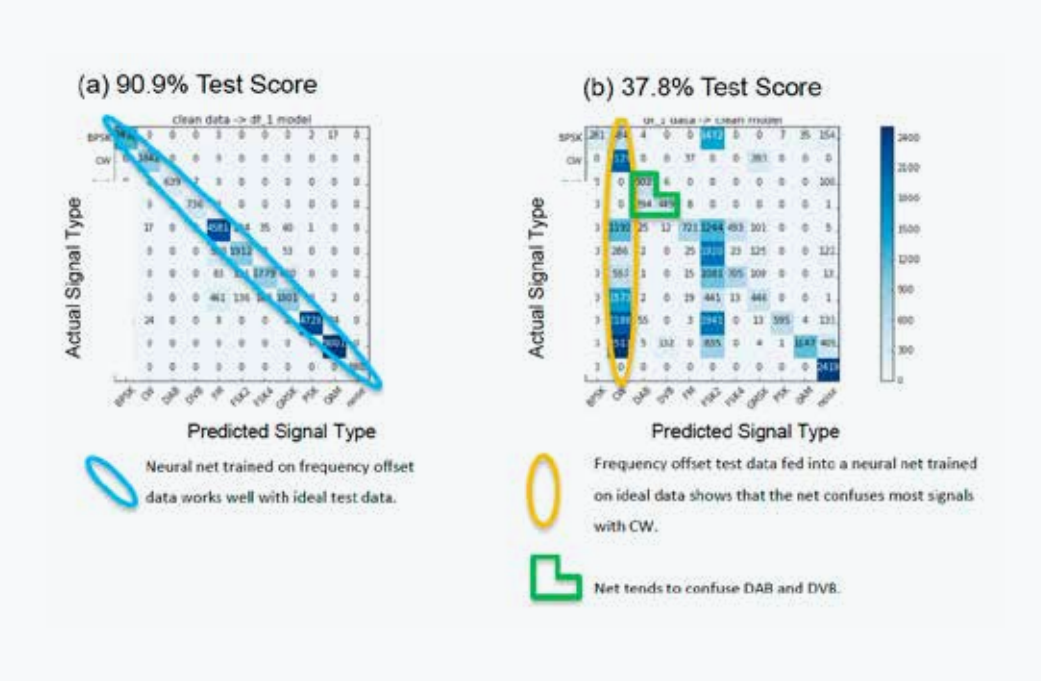
Table 1 *Overall results for different data sets*

Case	Training score	Test score	Comment
Clean signals	99.4%	99.4%	Ideal signal, high SNR
Frequency offset	98.1%	97.9%	Frequency offset, high SNR
Low SNR	97.5%	97.6%	Low SNR
Offset, low SNR	95.8%	95.7%	Combined

Table 1 shows the overall proportion of correct results in the data that was used to produce the confusion matrices in Figure 5. The method was tested on four different sets of data, with different noise and frequency offset characteristics. Signal classification was successful for all four data sets but, not surprisingly, it worked best with clean signals.

The results change significantly if we train the ANN on data from one set, then test it on data from another set. Figure 6 contains examples of the confusion matrices that we get if we do this. Matrix (a) shows the results of testing ideal data in an ANN that has been trained only on frequency offset data. This works fairly well, but the results are worse than with an ANN trained on the correct data set. Matrix (b) shows the effect of doing things the other way around, using frequency offset test data in an ANN trained on ideal data. The results here are poor. The net confuses most signals with CW, which makes sense because a frequency offset will manifest itself as a constant rate of phase change with time. The exception to this is that the net exhibits confusion between the multicarrier OFDM modulations of DAB and DVB.

These results reinforce the point that training sets need to include data obtained under a representative range of conditions.

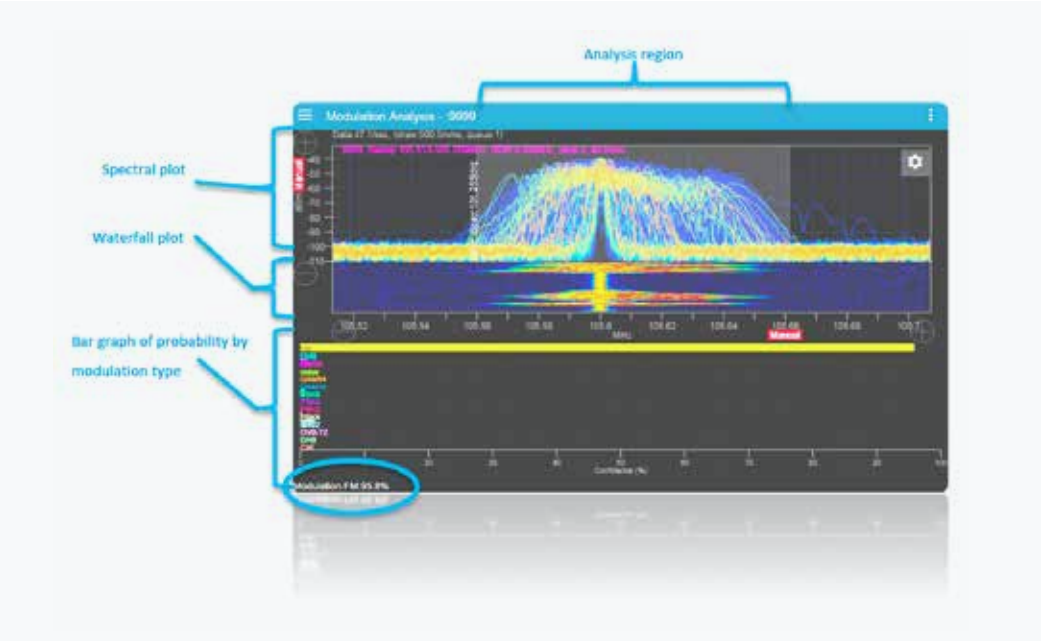


<< ***Figure 6*** *Confusion matrices for ANNs trained on the 'wrong' data*

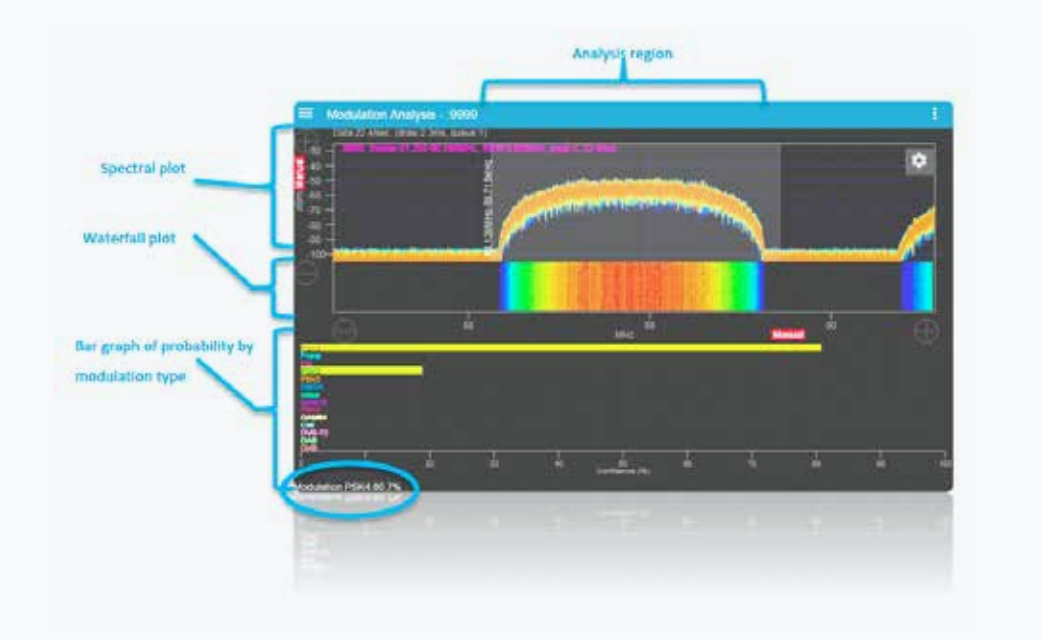
3.6 WHAT DOES IT LOOK LIKE IN REAL LIFE?

An ANN, trained for signal classification and running on an embedded sensor, was applied to live signals. Figure 7 and Figure 8 show the results achieved with an FM audio signal and a PSK4 signal. The FM signal was clearly identified as FM (95% probability), while the PSK4 was also identified correctly, but the classifier reported a smaller, but significant, probability that it might be PSK2.

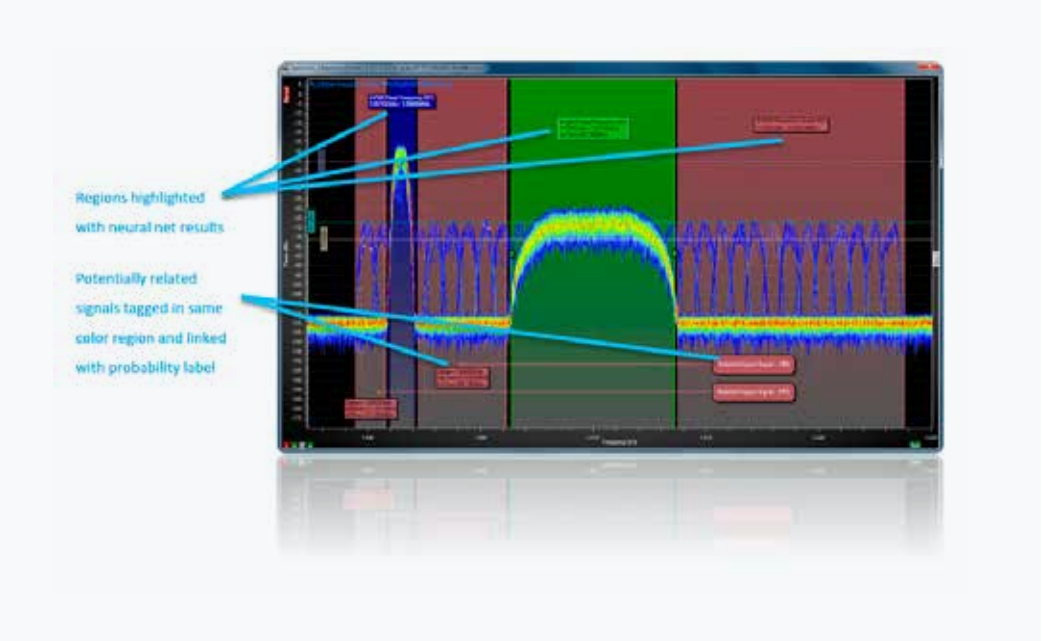
A very convenient way of reporting signal classification results to the user is to display them as color coded bars on a plot of the spectrum. Figure 9 shows a region of the spectrum around 1.01 GHz, displayed in this way, with 64 QAM, 4-PSK and 16 QAM frequency hopping signals clearly identified and highlighted.



<< ***Figure 7***
Classifier results for FM signal



<< ***Figure 8***
Classifier results for PSK4 signal



<< ***Figure 9***
Signal classification results displayed on a spectrum plot

3.7 WHY USE MACHINE LEARNING FOR SIGNAL CLASSIFICATION?

Signal classification by ML has several advantages over decision trees. The automation in the method allows it to be optimized for a set of signal types, without heavy engineer input, and to be improved easily by training with additional data. It is possible for the method to be implemented in a form that can be trained by the customer, if they do not want to reveal the training data to the vendor (for example, for reasons of security).

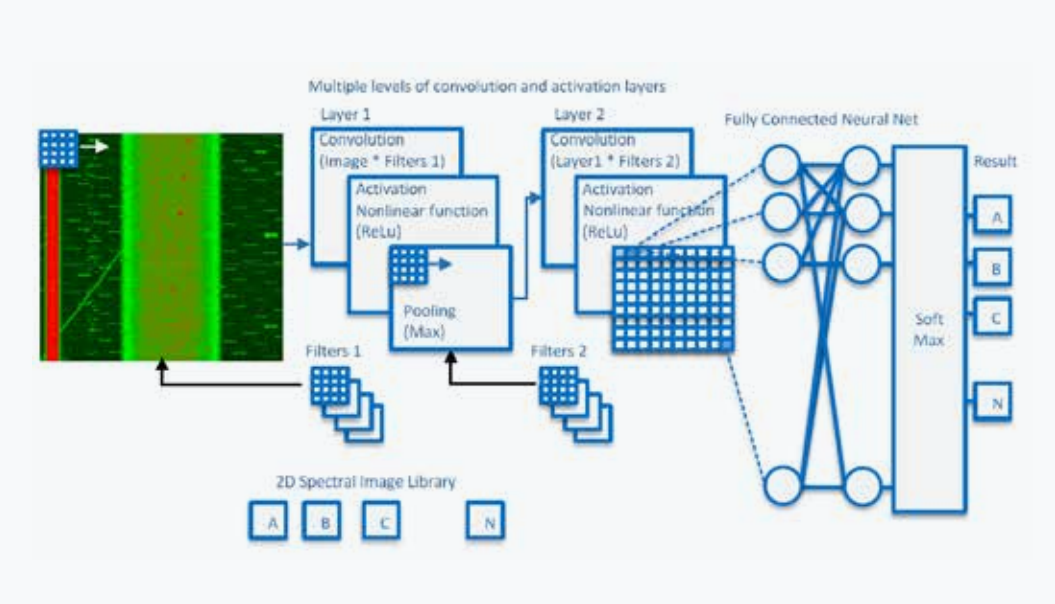
Another advantage of the ML implementation that we have outlined here is that, thanks to its nonlinearity, it may be able to fit the data more effectively than the linear DT method. The output from the ML also contains more helpful information, since it reports a probability for each signal type rather than just a single answer; this enables us to judge how confident we can be in the final answer.

4. MACHINE LEARNING FOR SIGNAL DETECTION

Another promising application of machine learning to RF spectrum intelligence extraction is in signal detection by means of pattern recognition.

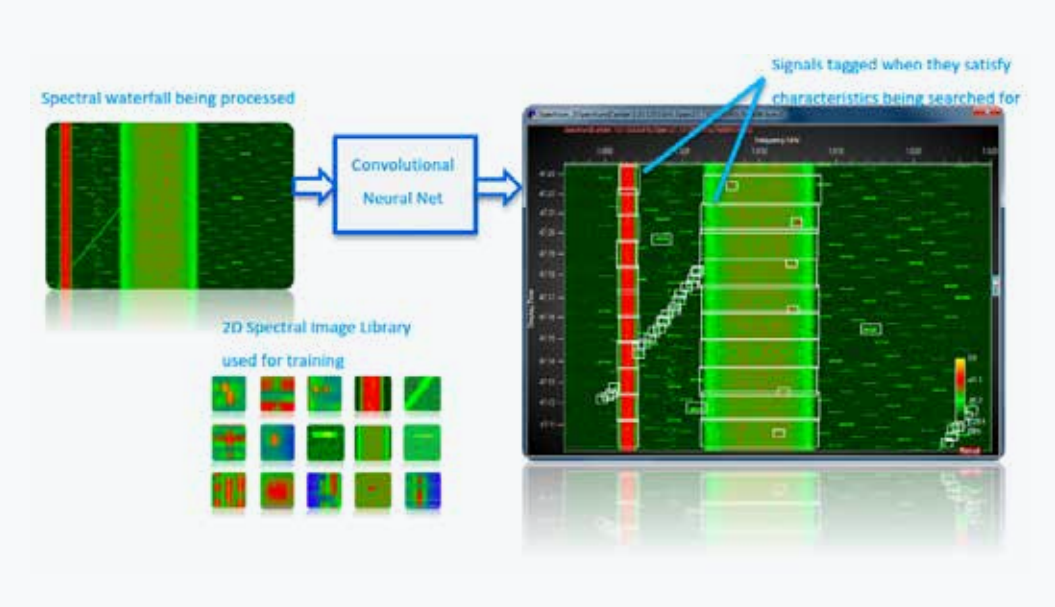
A 2D convolutional neural network (CNN) (Figure 10) is trained on a library of 2D spectral images (showing how the power depends on frequency and time, i.e., 'waterfall plot' images), each of which contains known features that might be found in a signal but are unlikely to be seen in noise (Figure 11). For example, there might be a burst of power that is constrained within hard boundaries in time and frequency, or that has a constant bandwidth but changes frequency at a constant rate.

This type of network was developed for image recognition and facial recognition. We are taking advantage of the fact that the system dynamics can be expressed as an image, so that the pattern recognition can follow the same principles as, for example, facial recognition. The 'features' learned by the CNN are essentially different from the 'features' used to train feedforward networks for signal classification; with the CNN, the features are elements of pattern that can be isolated in the power-frequency-time plot. While spectral characteristics can vary with frequency and time expansion or compression, the features will still be recognizable.



<< ***Figure 10***
Convolutional neural network for pattern recognition

The trained system will be able to detect which, if any, of the features are found in new live or archived data. This will provide an image signature for the IQ data, and the data can then be fed into a signal database for later signal classification analysis. Since the pattern recognizer can detect signals of types that it has not encountered before (provided that they display some of the signal-like features that it has learnt), it can act as a powerful data search engine for new signal types.



<< ***Figure 11***
Convolutional neural network searching for features in power-frequency-time plot

The task of recognizing signals has traditionally been performed by a human observer. The advantage of applying ML to this task is that, unlike a human observer, its performance will not deteriorate because of fatigue or distractions after long periods of observation. Therefore, ML may outperform a human in this role overall, even if it fails to do so in a single short test.

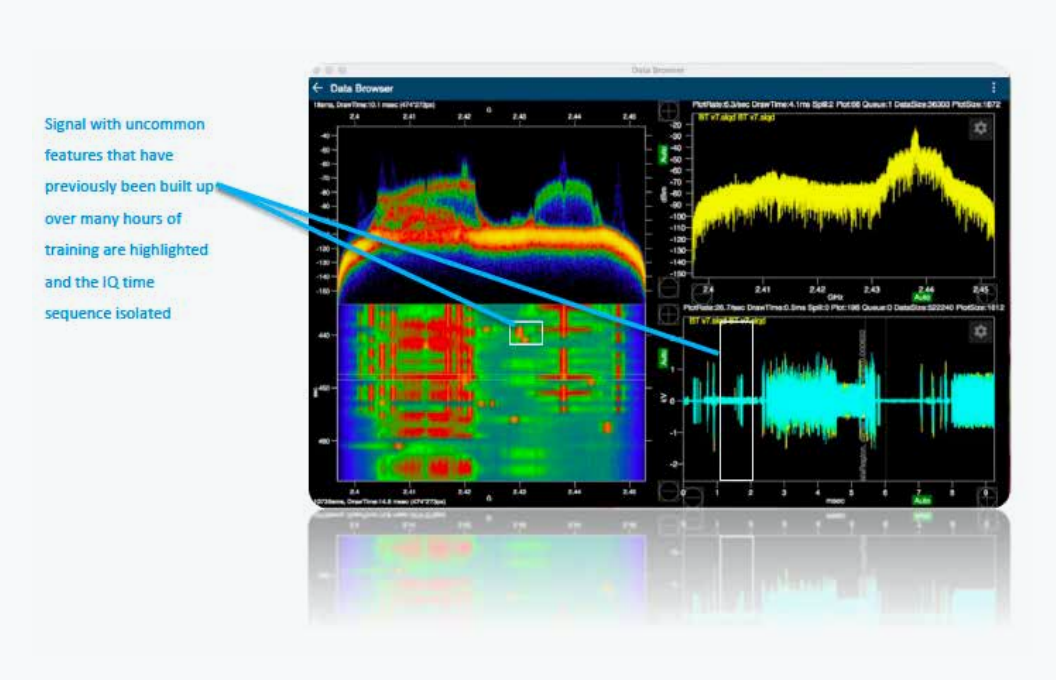
A further application of pattern recognition using CNNs is in RF fingerprinting of devices. In this case, the detailed characteristics of the patterns need to be considered. As an example, we can consider the signal emitted by a cellphone, which ramps up from zero power to the required value over some short interval of time. There will be an ideal theoretical form for the increase in power, but real phones are likely to deviate from the ideal in the shape and duration of the increase. Taken together, the precise behavior of all the applicable patterns forms the RF fingerprint of the device, which may allow individual phones to be identified.

5. MACHINE LEARNING FOR SPECTRUM MONITORING

It would be very useful to have a system that could automatically monitor the spectrum 24/7, without continuous human involvement.

To do this, it must learn the normal behavior of the spectrum, so that it can detect and report events that are 'interesting' or anomalous, and therefore worthy of special attention.

To accomplish this, we use the properties of convolutional neural networks, and also of unsupervised neural networks (clustering) to extract high level features from massive amounts of spectral data. An ANN is trained on hundreds of hours of normal spectral operating conditions, and the trained network is then able to recognize and highlight events that are uncommon or completely new (Figure 12).



<< ***Figure 12***
Example of spectrum monitoring using machine learning

6. EFFICIENCY OF ML PROCESSING

The analysis methods described in this white paper are applied to large data sets, so the efficiency of the processing is very important. Obviously, the ML analysis of spectrum data is useful only if it can be performed sufficiently fast: as a minimum, the real time analysis of live data needs to be able to keep up with the acquisition of data in real time.

A typical receiver with real time bandwidth of 100 MHz generates 500 MB/s of IQ data; the signal processing discussed here, running on a modest Intel i7, can process data at 1000 MB/s. ML analysis scales well if we increase the number of processors or the memory. For archive processing, the ultimate speed limit will be governed by the non-volatile memory read access time for local storage.

7. LOOKING TO THE FUTURE: FEATURE LEARNING

As we explained above, under our current method of applying ML to signal classification, the ANN operates with a set of features extracted from the captured IQ data, rather than using the data itself. This means that most of the information present in the IQ data is discarded. The set of features is selected with the aim of including the information that is most relevant to signal classification, in order that the discarded data should be less important. Therefore, we choose features that have led to successful signal classification by, for example, the DT method; the ML is aided by the insights of the engineer who chooses the features. The results presented in this white paper show that signals can be classified successfully using this feature-based approach. However, an alternative method is to supply the IQ data itself as the input to the ML. Under this 'Feature Learning' approach, the features will not be predetermined; they will be derived by the ML itself. The ML will no longer be helped by the insight of a human engineer, but it will also no longer be constrained by the limitations of that insight.

Using the IQ data directly will mean that the input to the ANN is a time series of complex rather than real data, and the architecture must also be adapted: the initial feature-analysis layers will be convolutional neural nets with complex weights.

This direct input mode can also be augmented by alternative signal representations - for example, wavelet or other transforms - which are also fed into convolutional nets. This is analogous to multi-sensor analysis, but all the inputs are derived from the same set of IQ data. The different representations will all contain the same information, but they will represent it in different ways, which may help the ML find features that can reliably distinguish between different types of signal.

8. CONCLUSION

We have successfully applied machine learning techniques to the automation of processes that are important for RF spectrum intelligence gathering.

In particular, a simple feedforward neural network has been trained to classify signals according to their modulation and protocol. Furthermore, we have shown how a convolutional neural network can detect signals, and potentially also fingerprint individual devices, by treating its input as an image and performing pattern recognition using a method that is analogous to image recognition or face recognition.

We have also seen how ML can be used to automate spectrum monitoring, which will be a major step forward in RF signal intelligence. All the applications that we have described have the potential for further improvement as we push towards the goal of reliable, automated spectrum monitoring and analysis.

LEGAL INFORMATION

Copyright © 2023 CRFS Limited

The copyright of this document is the property of CRFS Limited. Without the written consent of CRFS, given by contract or otherwise, this document must not be copied, reprinted or reproduced in any material form, either wholly or in part, and the contents of this document, or any methods or techniques available therefrom, must not be disclosed to any other person whatsoever.

CRFS Limited reserves the right to make changes to the specifications of the products detailed in this document at any time without notice and obligation to notify any person of such changes.

RFeye, CRFS and the CRFS logo are trademarks of CRFS Limited. All other trademarks are acknowledged and observed. Mention of third-party products does not constitute an endorsement or a recommendation.

All figures, data and specifications contained in this document are typical and must be specifically confirmed in writing by CRFS Limited before they apply to any tender, order or contract. CRFS takes every precaution to ensure that all information contained in this publication is factually correct but accepts no liability for any error or omission. No freedom to use patents or other property rights is implied by this document.

DISCLAIMER

The information contained in this document is not intended to amount to advice on which reliance should be placed. Although we take reasonable steps to ensure the accuracy of the information provided, we provide it without any guarantees, conditions or warranties as to its accuracy or reliability. To the extent permitted by law, we expressly exclude all conditions, warranties and other terms which might otherwise be implied by law. This document is subject to change without notice.

CONNECT WITH US

- Visit our website
- Send us a message
- Follow us on LinkedIn
- Watch our content on YouTube
- Browse our deployment library
- Read & subscribe to our blog

CRFS | EXTRAORDINARY **RF TECHNOLOGY**

CRFS is an RF technology specialist for the defense industry, national security agencies, and systems integration partners. We provide advanced capabilities for real-time spectrum monitoring, situational awareness, and electronic warfare support to help our customers understand and exploit the electromagnetic environment.

CRFS Inc
Chantilly, VA, USA
+1 571 321 5470
enquiries@crfs.com

CRFS Ltd
Cambridge, UK
+44 (0) 1223 859 500
enquiries@crfs.com

CRFS and RFeye are trademarks or registered trademarks of CRFS Limited. Copyright© 2023 CRFS Limited. All rights reserved. No part of this document may be reproduced or distributed in any manner without the prior written consent of CRFS. The information and statements provided in this document are for informational purposes only and are subject to change without notice.

bsi. ISO 9001:2015 Quality Management
UK Certificate number: FS576625