# AI-as-a-Service for Signal Processing

**Jeff Sieracki**, Engineering Director, Renesas Electronics

## Overview

Reality AI software from Renesas provides AI tools, optimized for solving problems related to sensors and signals. Working with acceleration, vibration, sound, electrical, RF, and proprietary signal sets, Reality AI Tools™ identify signatures of real-world events or conditions in sensor and signal data, and create software that can then spot those signals in the wild, notifying applications and devices so they can take action. Reality AI software also has uses for certain use cases with image, LiDAR, radar other image based applications. This white paper will cover technical aspects of our approach to machine learning and the architecture of our solution.

## Classifiers and Detectors

Reality AI Tools are machine learning tools that create classifiers, predictors and anomaly detectors – executable code that takes incoming observations and determine what has occurred in the real world based on those observations.

### Making Classifiers, Predictors and Detectors

Making classifiers and detectors involves determining which features of incoming signals are relevant to the classification/detection problem, and developing the means to identify which combination of features correspond to which results. Classifiers, predictors and detectors can be created through many different means, including traditional signal processing, statistical methods, or through machine learning.

Reality AI Tools use advanced machine learning techniques (described below), optimized for complex signal recognition.


© 2024 Renesas Electronics

Figure 1 - Classifier Example: An exercise app running on a smartwatch contains classifier/detector code that reads incoming accelerometer/gyroscope data, determines that an exercise has been completed, and determines which exercise it was.

## Using Classifiers, Predictors and Detectors

Once made, classifier, predictor or detector code can process new observations and deliver results. Results can be in the form of binary determinations, type classifications, or predictions/estimates of quantitative measures, depending on how it was constructed.

# Signals are Different

## What are Signals?

Signals are sensed-phenomena originating in the physical world, and when digitally sampled contain an umber of characteristics that make them fundamentally different from other kinds of time-series and spatial data, like machine logs, discrete event data and locations data.

Signals data include sound, images, accelerometry, vibrations, electrical, RF and inputs from other sensors covering a wide range of phenomena.

## What Makes Signals Different?

**Signals are often characterized by:**

**HIGH SAMPLE RATE**

High number of samples per unit time/space relative to other kinds of data (such as event logs, discrete time series, or geolocation data). This gives non-signal tools great difficulty because important distinguishing features are obscured by high-dimensionality and mathematically complex relationships.

**JITTER & PHASESHIFTS**

Irregular variation and time-shifting of target signatures in data due to physical phenomena, as well as inexact relationship between observed events and measured signals.

 © 2024 Renesas Electronics

### DYNAMIC NOISE & DISTORTION

Obscuring elements in collected signals that result from the physics of the sensors, and variation or other natural phenomena in the collection environment.

### TRANSIENT SIGNATURES

Brief and non-stationary targets in signals data that can be very hard to separate from noise and other aggregated signals, but which often contain the most revealing signatures. Local or transient features may be just as important as more widespread, slower, or persistent signal features; best results come when both are considered simultaneously.

These aspects and others make signals data very difficult to analyze using statistical or machine learning tools intended for discrete time-series or categorical data.

## Working with Signals the Traditional Approach

Engineers, however, have developed a number of tools for making signal data more tractable for statistical analysis. Signal processing typically begins by subjecting the incoming signal to a Fast Fourier Transform (FFT), filter banks, or a linear systems analysis to find the amount of energy in different parts of the signal's frequency and time domains.

This is the way signal analysis has been taught in engineering schools since the 1970s, and there are a number of robust tools supporting this approach – perhaps the most widely used being the Matlab Signal Processing Toolbox[1]. The output of these processes can then be treated as features in statistical detection methods, but rely heavily on the engineer's understanding of the physics of the connection between the signal and what she is trying to detect.

Figure 2 - Sound of ocean waves crashing into rocks

Figure 3 - Vibrations of a bearing in an industrial machine

Figure 4 - Images of a cell-tower collected by drone. Reality AI has highlighted rust in purple

[1] Matlab Signal Processing Toolbox is a trademark of The MathWorks, Inc


© 2024 Renesas Electronics

# Machine Learning for Signals

## Traditional Machine Learning Doesn't Work Well on Signals

For years now machine learning has posed an obvious alternative to these traditional methods of analyzing signals data. But for all the reasons above, signals data confounds most machine learning tools. The high dimensionality of signal analysis problems and complex mathematical relationship between the sampled values and underlying features leads to long processing times and low accuracy for most real world problems.

## It's the Features!

At the heart of the matter is feature discovery. In machine learning parlance, features are the specific variables that are used as input to an algorithm. Features can be selections of raw values from input data, or can be values derived from that data. With the right features, almost any machine learning algorithm will find what you're looking for. Without good features, none will – and that's especially true for real-world problems where data comes with lots of inherent noise and variation.

When signals become complex enough so that the individual values recorded no longer work as features, most engineers turn first to the traditional RMS energy, peak-to-peak measures, or the FFT to create the features – treating the averages or FFT coefficients as the relevant features for training. But the problem is that these features discard much of the information necessary to make the judgements that you most want to make. RMS can certainly help identify some conditions, and an FFT run with relatively high frequency and time resolution can be quite valuable. But high-level descriptive statistics discard almost all the essential signature information you need to find granular events and conditions that justify the value of the sensor implementation in the first place. And as this excellent example from another domain shows, descriptive statistics just don't let you see the most interesting things:

Figure 5 - Why basic statistics are never enough. All of these plots have the same X and Y means, the same X and Y standard deviations, and the same X:Y correlation. Without the ability consider all of the source detail, your algorithm would never see any of these patterns in your data.

Source: https://www.autodeskresearch.com/publications/samestats


© 2024 Renesas Electronics

Instead of relying on aggregate features defined by the traditional signal processing tools, Reality AI discovers features directly from the raw sensor signal data in all of its inherent noisiness and complexity. This approach allows for consideration of information in both time and frequency domains, and allows for detection of conditions with much more subtle signatures.

Figure 6 - This example shows a time series of data from an accelerometer attached to a machine in a manufacturing facility. X, Y and Z components of the acceleration vector are averaged over one second.

There is very little information in this data – in fact, just about all it can tell us is which direction is gravity. This data was provided from an actual customer implementation in a manufacturing facility, and is basically useless for anomaly detection, condition monitoring, or predictive maintenance.

Figure 7 – This example shows vibration data pre-processed thru a Fast Fourier Transform (FFT) at high frequency resolution. The X-axis is frequency and the Y-axis is intensity. This data is much more useful than Figure 6 – the spikes occurring at multiples of the equipment's base rotation frequency give important information about what's happening in the machine and is most useful for rotating equipment. FFT data can be good for many applications, but it discards a great deal of information from the time-domain. It only shows a single snapshot in time – this entire chart is an expansion of a single data point from Figure 6.

 © 2024 Renesas Electronics

Figure 8 – Raw time-waveform data as sampled directly from the accelerometer. This data is information-dense – being the raw data from which both the simple averages in Figure 6 and the FFT in Figure 7 were computed. Here we have extremely detailed frequency information, coupled with important tine information such as transients and phase. We also have all of the noise, however, which makes this kind of data very difficult for human analysts to use directly. But a data-driven feature discovery algorithm like Reality AI can extract maximum value from this kind of data, creating a custom transform that zeros in on exactly those parts of the signal – and only those parts of the signal – that are relevant for your detection problem.

## Data-Driven Feature Discovery

Our algorithms are based on patented techniques using mathematics described in the literature as sparse decomposition or sparse coding, as well as a number of other related proprietary techniques. These techniques operate unsupervised or semi-supervised to dynamically discover an optimal feature set for describing similarities between signals of the same class and differences between signals of different classes.

When using Reality AI Tools, we will encourage you will point the algorithms at your source data in its most detailed, un-preprocessed form, and run a process that we call AI Explore™. AI Explore will then attempt to identify the features that best separate your training classes (if your data is labeled), or that best characterizes “normal” (for unlabeled anomaly detection). It will then offer a selection of machine learning models based on the best features – ranking them by accuracy and by computational complexity. Those models can then be trained on whatever data is available, and validated using holdout sets or a variety of statistical techniques.

Figure 9 - Using a Fourier transform to define features in complex, dynamic signals, there is loss of information due to the transform, noise and jitter.

Figure 10 - Our methods consolidate features and de-blurr jittered information. Features are much more precise, and more easily learned.

 © 2024 Renesas Electronics

Once the right features have been identified, the classification/detection problem becomes much simpler and even basic machine learning algorithms become much more effective– delivering more accurate results with less training data.

This approach to data-driven feature discovery is the core insight that drives Reality AI technology and its unique effectiveness with signal data

# Reality AI vs Deep Learning

One other type of machine learning approach that has recently shown good results on a number of different signal-related problems is Deep Learning. Deep Learning is a branch of machine learning based on a set of algorithms that attempt to model high-level abstractions in data by using multiple processing layers with complex structures – usually composed of recurrent or convolutional neural networks. Deep Learning also uses an approach for data-driven feature discovery – essentially using lower layers of convolutional neural networks or similar algorithms to find features that higher layers can then learn from. Deep Learning has been used to good effect for image sorting and tagging, image and scene interpretation, automatic speech recognition, and natural language processing.

## Good at Different Things

While there is certainly overlap between problems for which Deep Learning is well suited and problems that Reality AI can address, there are also some differences. In images, for example, Deep Learning is particularly good at identifying objects and characterizing scenes, while Reality AI is particularly good at problems related to subtlety in texture: identifying surfaces and spotting surface discontinuities or nuanced anomalies. You can see this surface classification and discontinuity identification aspect most readily in image-related data, but that same approach is at work with signals as well, for example in identifying subtle variations in vibration or machine noise to spot maintenance conditions in complex machinery.

### The Right Tool for the Right Job Examples:

| | **Deep Learning** | **Reality AI** |
|---|---|---|
| UAV/drone | Asset surveys identifying dump trucks and bulldozers | Surface classifications identifying characteristics of microterrain. |
| Sound | Understanding speech | Understanding machine noise<br>Complex anomaly detection |
| Images | Identifying objects<br><br>Describing scenes | Texture and related nuance<br><br>Subtle detection in noise or at limits of resolution |

 © 2024 Renesas Electronics

## Easier Setup, Less Training Data and Embedded Deployment

But even when both Deep Learning and Reality AI are capable of delivering good results, Reality AI has some important advantages:

| | Deep Learning | Reality AI |
|---|---|---|
| Setup Time | Each new problem requires a new model to be designed from scratch, leading to long lead-times and upfront costs | Configuration in hours. |
| Data requirements | Large amounts of training data required for most every problem.<br>Multiple layers of neural networks with lots of nodes each requiring sufficient data for statistically significant training. | Much less training data required - often several orders of magnitude less. |
| Training times | Weeks to months | Hours to days |
| Hardware requirements | Often requires specialized hardware to deliver required processing power - both for training and in field deployment. | No specialized hardware required |
| Embedded deployment | Possible on specialized hardware optimized for deep learning and carrying significant power requirements. | OK for microcontroller or DSP-driven environments.<br>Suitable for cycle, memory- and power-constrained deployment.<br>Suitable for real-time detection at the edge |

# Technical Architecture

### Customers interact with Reality AI thru two methods:

### Reality AI Tools

Web-based interface to tools for data curation, classifier maintenance, retraining, testing and validation, and API dashboard monitoring.

### Reality AI API

Both Reality AI Tools and the Reality AI API are front-ends to a multi-tiered server environment that handles all transactions.

Customers who are using Reality AI Tools to create embedded classifiers will also have interaction methods specific to their deployment.

 © 2024 Renesas Electronics

## Cloud Deployment Architecture

Both Reality AI Tools and the Reality AI API are front-ends to a multi-tiered server environment that handles all transactions.

Figure 5 - Reality AI Technical Architecture for Cloud Deployments Multi-tiered server architecture for scalable processing and load-balancing

## Cloud Deployment Scalability

Reality AI relies on a multi-tiered, load-sensing, load-balancing server architecture that determines when additional capacity is needed and then provides it:

### Reality API Management

Handles API security and determines need for additional Process Servers to handle incoming volume. We employ a RESTful API design to enable scalable, asynchronous processing for our customers.

### Process Servers

Manages incoming API transactions from Reality AI Tools and from customer API usage. Each incoming API call, once validated, is then farmed out to Math Servers and Business Servers as appropriate. When results are returned, they are combined into a response and passed back to the requesting application. Monitors Math Servers and Business Servers and provisions additional capacity as needed.

### Math Servers

Interfaces with other systems at Reality AI software from Renesas, and handles both administrative and economic transactions (eg usage tracking).


© 2024 Renesas Electronics

### Business Servers

Interfaces with other systems at Reality AI software from Renesas, and handles both administrative and economic transactions (eg usage tracking).

As additional capacity is needed, each service tier is able to provision and spin up additional servers in the tier below, allowing services to scale seamlessly and without performance degradation.

## Embedded Deployment

For customers requiring embedded deployment, code that would have run on a Math Server in the cloud is instead exported in a form suitable for adaptation to the specific embedded target. Typically, this is in the form of compiled code, though other options are also available. Contact us for more information.

# Security

Reality AI takes security seriously, and our systems are engineered for privacy and security from the ground up.

## Standard Security Measures

Our severs our secured by industry standard security measures, including SSL encryption of all IP traffic and web transactions. Customers are provided unique administrative log-in credentials, and API transactions handled using security tokens generated on a dynamic basis.

In addition, we limit public address exposure of our machines to monitored API portals

## Data Security

Customers can choose to securely upload and store data with us in a persistent data store. This is maintained separately for each user, and data is not shared between or aggregated across customers.

Customers may choose to link data dynamically from other data sources. For example, they may furnish us unique credentials to an Amazon AWS bucket, which we will access on a transient basis only when needed for processing. We can also support links to private servers, so that your data remains in customer control up to the moment it is processed. Unless otherwise required by the customer’s use case, Reality AI discards all processed observations.

Finally, for sensor data being shown to a deployed, production classifier/detector where that data is sensitive and must be kept private, or where the data is too sensitive to transmit thru the cloud (some sound recordings, for example), it is possible to configure the system to complete pre-processing locally so that only feature vectors are transmitted for classification. In this way, features unrelated to the classification are not available for observation, and raw signals need not leave the local device.

 © 2024 Renesas Electronics

## Anonymity and Data Sanitization

Because our algorithms are sensor-agnostic and 100% data-driven, it is entirely possible for customers to use our technology to create classifiers/detectors without disclosing any specifics about the data at all, and using completely anonymized labeling. This functionality is particularly attractive to customers working with personally identifiable information, or those with proprietary sensors and sensor configurations who do not wish to disclose details of the data collection methods or pre-processing.

# Conclusion

Reality AI software from Renesas provides AI tools with real-time analytics optimized for solving problems related to sensors and signals. Working with non-visual sensing and proprietary signal sets, the software identifies signatures of real-world events or conditions in the sensor and signal data so users can create models to deploy on their systems to identify signals and notify applications and devices to take action. This Reality AI software running on Renesas processors helps deliver endpoint intelligence to end products and support solutions across all markets.

# Resources

- renesas.com/realityai

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.
These resources are intended for developers skilled in the art designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only for development of an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising out of your use of these resources. Renesas’ products are provided only subject to Renesas’ Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.
(Rev.1.0 Mar 2020)

**Corporate Headquarters**
TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan
https://www.renesas.com

**Contact Information**
For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
https://www.renesas.com/contact-us

**Trademarks**
Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Renesas Electronics Corporation
renesas.com

Doc Number: R30WP0006EU0100

 © 2024 Renesas Electronics