INL/EXT-18-51429
Revision 0

Light Water Reactor Sustainability Program

Performance of Advanced Signal Processing and Pattern Recognition Algorithms Using Raw Data from Ultrasonic Guided Waves and Fiber Optic Transducers

September 2018

U.S. Department of Energy

Office of Nuclear Energy

INL/EXT-18-51429
Revision 0

Light Water Reactor Sustainability Program

Performance of Advanced Signal Processing and Pattern Recognition Algorithms Using Raw Data from Ultrasonic Guided Waves and Fiber Optic Transducers

Andrei Gribok, Kevin P. Chen, Zhi-Hong Mao

September 2018

Prepared for the
U.S. Department of Energy
Office of Nuclear Energy

Light Water Reactor Sustainability Program

Performance of Advanced Signal Processing and Pattern Recognition Algorithms Using Raw Data from Ultrasonic Guided Waves and Fiber Optic Transducers

INL/EXT-18-51429
Revision 0

September 2018

Approved by:

Name Title [optional]	Date
Name Title [optional]	Date
Name Title [optional]	Date
Name Title [optional]	Date

ABSTRACT

The Light Water Reactor Sustainability Program was initiated to evaluate technologies that could be used to perform online monitoring of piping and other secondary system structural components in commercial nuclear power plants. These online monitoring systems have the potential to identify when a more detailed inspection is needed using real-time measurements, rather than at a pre-determined inspection interval.

This transition to condition-based, risk-informed automated maintenance will contribute to a significant reduction of operations and maintenance costs that account for most nuclear power generation costs.

This report describes the current state of research related to ultrasonic-guided wave testing and its application to detecting defects in commercial nuclear power plants. The report analyzes the applicability of the guided wave technology to secondary piping systems, as well as studying the potential for expanding the range of guided wave technology to include bent piping and other piping components. The ultrasonic-guided waves can inspect long stretches of straight piping; however, more complex geometries such as elbows, welds, and tees are causing spurious reflections and coherent noise, which significantly decreases the sensitivity of the technique. To deal with these limitations, high-definition fiber optic sensors are applied to complex piping geometries, and advanced machine learning algorithms are used to detect deviations from healthy states.

This report also analyzes the performance of advanced signal processing and machine learning-based pattern recognition algorithms in detecting defects in secondary structures. It is demonstrated on guided wave data collected at nuclear power plants that the independent component analysis can separate different coherent noise components and segregate them from useful signals. It also demonstrates that advanced machine learning techniques, such as neural networks, support vector machines, and deep learning networks, can detect minor defects present in inspected structures.

Recommendation about the applicability of advanced machine learning techniques to online piping monitoring are also given.

SUMMARY

This report describes the performance of advanced signal processing and machine learning techniques to detect corrosion degradation in secondary structures using ultrasonic-guided wave testing and high-resolution fiber optic in nuclear power plants (NPPs), which is being conducted under the United States (U.S.) Department of Energy's (DOE's) Light Water Reactor Sustainability (LWRS) Program.

The LWRS Program, funded by DOE's Office of Nuclear Energy (DOE-NE), aims to provide scientific, engineering, and technological foundations for extending the life of operating light water reactors (LWRs). This program involves several goals, one of which is ensuring the safe operation of the passive components in NPPs, such as concrete, piping, steam generators, heat exchangers, and cabling.

Within the LWRS Program, the Advanced Instrumentation, Information, and Control (II&C) Systems Technologies Pathway conducts targeted research and development (R&D) to address aging and reliability concerns with the legacy analog instrumentation systems, structures, and components (SSCs) and related information systems of the operating U.S. LWR fleet. This work involves two major goals: (1) ensuring legacy analog II&C systems are not life-limiting issues for the LWR fleet, and (2) implementing digital II&C technology in a manner that enables broad innovation and business improvement in the NPP operating model. Resolving long-term operational concerns with II&C systems contributes to the long-term sustainability of the LWR fleet, which is vital to the nation's energy and environmental security.

The goals of the LWRS Program are addressed through a number of pilot projects that target realistic opportunities for increasing sustainability, safety, and economic efficiency of the existing NPP fleet. It is generally recognized that the biggest challenge for existing NPPs is economic viability. Reducing operations and management costs are one of the most pressing problems facing the nuclear power generation industry. Operations and maintenance costs comprise approximately 60–70% of the overall generating cost in legacy NPPs. Only 15–30% of the costs are attributed to obtaining and producing the fuel.

Furthermore, of the operations and maintenance costs in U.S. plants, approximately 80% are labor costs. To address the issue of rising operating costs and economic viability, companies that operate the national nuclear energy fleet started the "Delivering the Nuclear Promise Initiative" in 2016, which is a three-year program aimed at maintaining operational focus, increasing value, and improving efficiency.

ACKNOWLEDGEMENTS

The authors would like to thank Heather Feldman and Kurt Crytzer of the Electric Power Research Institute (EPRI) for their help and leadership in organizing the joint workshop on structural health monitoring and for providing technical guidance and feedback during the development of the joint research plan. The author is also grateful to Kenneth Thomas of Idaho National Laboratory (INL) for his insights and encouragements during technical discussions.

The authors would also like to extend their special thanks to Alan Puchot of Southwest Research Institute (SwRI) for providing reports, software, and guided wave data.

CONTENTS

FIGURES

TABLES

ACRONYMS

ϕ-OTDR	phase-sensitive optical time-domain reflectometry
AOM	acoustic-optical-modulator
CNN	Convolutional Neural Network
DAS	distributed acoustic sensing
DNN	deep neural network
DOE	U.S. Department of Energy
DOE-NE	U.S. Department of Energy–Office of Nuclear Energy
DWM	dispersive wave mode
EDFA	Erbium-doped fiber amplifier
EPRI	Electric Power Research Institute
FC	fiber coupler
FFT	Fast Fourier transform
FRM	faraday rotator mirror
FW-PHM	Fleet-Wide Prognostics and Health Management
GCV	Generalized Cross Validation
GW	guided wave
IC	independent component
ICA	independent component analysis
ICOMPRPS	information complexity regularization parameter selection method
ID	inner diameter
II&C	advanced instrumentation, information, and control
LM	Levenberg-Marquardt algorithm
LWR	light water reactor
LWRS	Light Water Reactor Sustainability
MDL	minimum description length
MDP	Morozov's Discrepancy Principle
ML	machine learning
MLP	multilayer perceptron
MSE	Mean Squared Error
MsS	magnetostrictive system
NDE	non-destructive examination
NN	neural network
NPP	nuclear power plant

OFDR	Optical Frequency Domain Reflectometry
OLS	ordinary least squares
PAC	probably approximately correct
PBS	polarizing beam splitter
PC	polarization controller
PCA	principal component analysis
PSD	power spectral density
PZT	lead zirconate titanate
R&D	research and development
RBF	radial basis function
ReLU	rectified linear unit
RIC	Regularization Information Criterion
RPSM	regularization parameter selection method
SGD	Stochastic Gradient Decent
SH	shear horizontal
SNR	signal-to-noise ratio
SSC	system, structure, and component
SVM	support vector machine
SwRI	Southwest Research Institute
TLS	tunable laser source
U.S.	United States
UGW	ultrasonic-guided wave
URE	Unbiased Risk Estimator
UT	ultrasonic testing

1. ULTRASONIC-GUIDED WAVE TESTING

The United States (U.S.) Department of Energy's (DOE's) Light Water Reactor Sustainability (LWRS) Program, funded by DOE's Office of Nuclear Energy (DOE-NE), aims to provide scientific, engineering, and technological foundations for extending the life of operating light water reactors (LWRs). This program involves several goals, one of which is ensuring the safe operation of the passive components in nuclear power plants (NPPs), such as concrete, piping, steam generators, heat exchangers, and cabling.

Within the LWRS Program, the Advanced Instrumentation, Information, and Control (II&C) Systems Technologies Pathway conducts targeted research and development (R&D) to address aging and reliability concerns with the legacy analog instrumentation systems, structures, and components (SSCs) and related information systems of the operating U.S. LWR fleet. This work involves two major goals: (1) ensuring legacy analog II&C systems are not life-limiting issues for the LWR fleet, and (2) implementing digital II&C technology in a manner that enables broad innovation and business improvement in the NPP operating model. Resolving long-term operational concerns with II&C systems contributes to the long-term sustainability of the LWR fleet, which is vital to the nation's energy and environmental security.

This report describes the application of independent component analysis (ICA) and machine learning (ML)-based advanced pattern recognition techniques to detect corrosion-induced defects in commercial NPPs and analyzes the applicability and benefits of ICA and ML techniques when applied to guided wave (UGW) technology and fiber optic sensors to detect corrosion in secondary circuits, as well as studying the potential for expanding the range of UGW technology to include complex geometries and piping components using fiber optic sensors. UGWs can inspect long stretches of straight piping; however, more complex geometries that include elbows, welds, and tees can cause spurious reflections and coherent noise, which significantly decreases the sensitivity of UGW systems. The potential of ICA and ML to improve detection sensitivity is analyzed and practical recommendations are provided. It is demonstrated on UGW data collected at a commercial NPP that ICA, under certain conditions, can separate different coherent noise components and has the potential for improving signal-to-noise ratio. Different advanced pattern recognition techniques are applied to UGW and fiber optic sensors to evaluate their ability to detect different types of defects in secondary piping.

The transition to condition-based, risk-informed automated maintenance will contribute to a significant reduction in operations and maintenance costs accounting for approximately 66% of the total operating cost of NPPs. These costs are significantly higher in comparison to gas (13%) and coal plants (22%) [1]. To deal with the issue of increasing operating costs and economic sustainability, companies

that operate the national nuclear energy fleet started the "Delivering the Nuclear Promise Initiative" [2] in 2016, which is a three-year program aimed at maintaining operational focus, increasing value, and improving efficiency. While stressing the industry's commitment to safety and security, the initiative also emphasized economic viability and competitiveness in the current deregulated energy market.

As a means of defect detection technology, UGW testing has been successfully implemented in the field of non-destructive examination (NDE) for several years now [3,4,5]. The velocity of the guided waves (GWs) is directly dependent on the thickness of the material, which is characterized by the dispersion behavior of the modes of the GWs. Hence, the difference in the thickness of the component will give a variation in the time of GW arrival, which forms the physics foundation of UGW detection and monitoring systems.

Tests conducted by the Imperial College in London [5] have proved the applicability of the utilization of shear horizontal (SH) waves for calculating the average thickness of a plate along a line between two transducers. This work also helped generate a methodology for extracting the value of the thickness measured by SH GW signals, which were characterized by temperature. However, application of GWs to the piping systems of NPPs face additional serious challenges, such as complex geometric shapes, hostile environments, and insulation.

The piping system is one of the most valuable assets in legacy NPPs, with inspections performed on a regular basis. The technical basis for the inspection period could be based on analytical predictive analysis, plant operating experience, industry experience, susceptibility of the equipment, engineering judgment, and acoustical or vibration measurements. However, because of the significant length of the piping systems, the problem of identifying specific piping components that need to be inspected during an outage remains a challenge. Thus, many unnecessary inspections are performed, which adds to planned downtime and lost revenue. The well-established technology of UGWs offers new possibilities in the inspection of large portions of piping systems with few sensors. UGWs are mechanical or elastic waves that propagate at low frequencies—either sonically or ultrasonically through the walls of a pipe—and are bounded and guided by those walls. The velocity and wave modes of GWs are strongly influenced by the geometry of the guiding boundaries. In the pipe, the UGWs exist in three different wave modes—longitudinal, torsional, and flexural. Because the UGWs are mechanical waves, they are generated either through piezoelectric or magnetostrictive transducers that convert electrical magnetic fields into mechanical energy. Once the mechanical wave is generated with a set of piezoelectric or magnetostrictive sensors arranged in a collar around the pipe, it is transmitted through the walls of the pipe and reflected back from any discontinuities (e.g., flaws) of the surface of the wall, as shown in Figure 1.

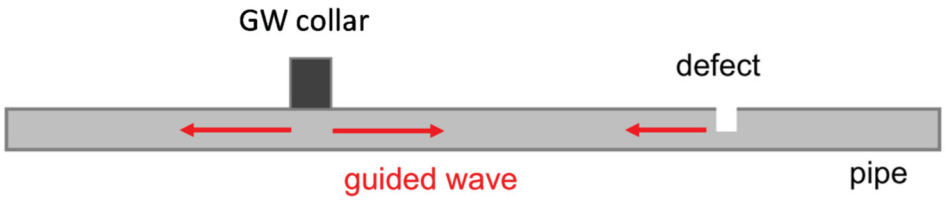

Figure 1. The principle of GW corrosion monitoring (figure source [4]).

UGW inspection has numerous advantages over other NDE techniques, which include:

- the ability to inspect large sections of piping with a single sweep
- the ability to inspect inaccessible locations
- the flexibility of its sensors to be mounted permanently
- the option that it may be used for inspection while the system is operating
- the ability to inspect pipes from 2–96-in. diameter.

The main advantage of GW inspections over conventional ultrasonic testing (UT) inspections is shown in Figure 2. In contrast to conventional UT inspections, GW technology can cover tens of meters in one inspection session. Traditional UT inspections are highly localized and can only detect flaws within the proximity of the sensor location.

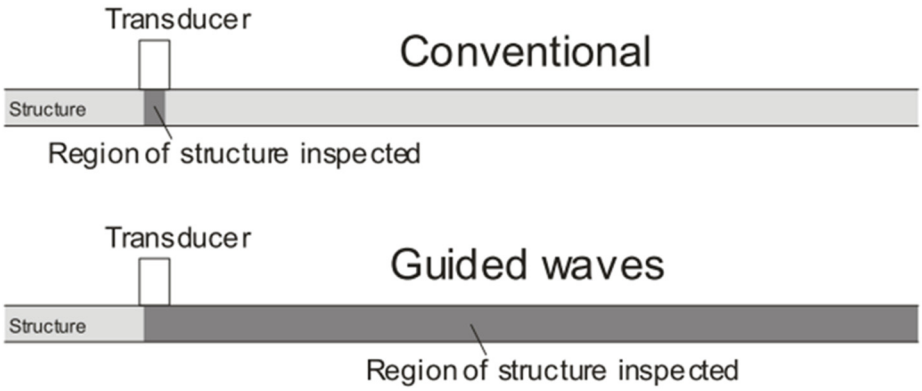

Figure 2. Differences between GW inspections and conventional UT inspections (figure source [4]).

Despite these benefits, GW technology is challenged when applied to power plants in general and NPPs in particular. Piping systems in electric power plants come in various configurations and geometries; for example, they have thousands of elbows, bends, tees, valves, and flanges. These geometries are not a friendly media for GWs. Geometries other than strait pipe attenuate and distort GWs, thereby making inspections beyond them difficult. Also, while being a perfect tool for locating the damage in pipes, GWs cannot determine the size of the flaw with acceptable accuracy. In summary, provided the GW technology can overcome the limitations of complex geometries, it is the perfect tool for answering the “where to inspect?” question.

2. ICA APPLICATIONS TO GW SIGNALS COLLECTED ON A HEAT EXCHANGER SHELL

As a mean to detect defects, GW testing has been successfully implemented in the NDE field [6]. The major advantage of this technology is its ability to cover large stretches of metal structures from a single location. The detection range of a single sensor location extends a few hundred feet in both directions from the sensor location.

The large detection coverage also makes it economically viable to permanently install UGW systems for continuous online monitoring. The UGW systems are traditionally applied as inspection tools where the system is installed, measurements are taken, and the system is moved to a different location. Recent developments in technology, however, allow permanent installation, thus significantly reducing logistics of the inspections eliminating insulation removal and scaffolding.

The UGWs are mechanical waves, which are generated either through piezoelectric or magnetostrictive transducers converting electrical magnetic fields into mechanical energy. Magnetostrictive materials are able to convert the magnetic field into kinetic energy or mechanical stress, while piezoelectric materials convert the electric field into mechanical stress or mechanical wave. For wave generation, the magnetostrictive method relies on the Joule effect [7], while for wave detection, it relies on the Villari effect [7].

Both techniques have their advantages and disadvantages and are widely used to generate mechanical waves for the purpose of off-line NDE inspections. Both approaches are capable of generating low-frequency (10–100 kHz) waves, which include an audible range to avoid attenuation during propagation though inspected structures. Both methods generate one of three types of GWs: torsional, longitudinal, and flexural modes.

These three modes interact with the discontinuities in a metallic structure and are reflected back to the sensor from those discontinuities, thus pinpointing the location of structural degradation. The longitudinal and torsional waves are the most widely used for NDE due to their sensitivity to defects. Longitudinal waves propagate along the object with compression and rarefication, while torsional waves propagate as a result of a medium being twisted and released.

The data used in this report were obtained with a magnetostrictive system (MsS) developed at Southwest Research Institute (SwRI) [8,9]. The MsS structural health monitoring system has been developed at SwRI in collaboration with the Electric Power Research Institute (EPRI) to monitor corrosion in large secondary structures of NPPs [8,9]. The description of the MsS provided in this section follows descriptions presented in [8,9]. The block diagram of the MsS is shown in Figure 3. Two major

parts of the system are MsSs and an MsSR3030 unit. The unit works in pulse-echo mode by generating and receiving GWs through MsSs. The system has 17 sensors that are attached to the structure being tested (e.g., the heat exchanger shell).

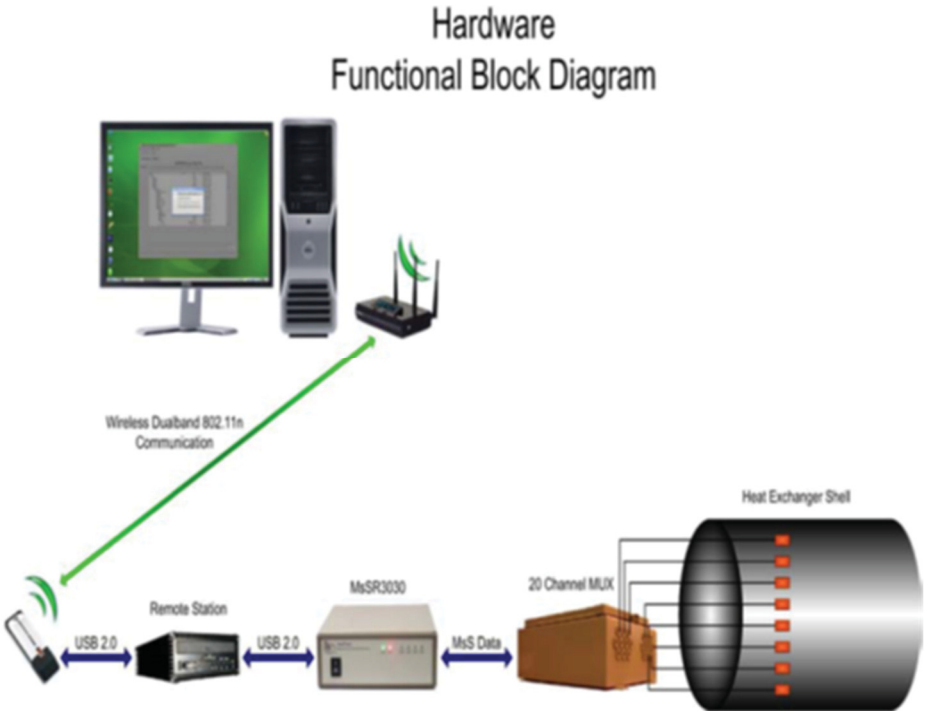

Figure 3. Block diagram of the MsS monitoring system (figure source [8]).

The system also has a multiplexer and control computer. The multiplexer makes it possible to connect multiple sensors to a single MsSR3030 unit. The MsS can generate and detect waves propagating in both directions from the sensor, and is designed to generate SH torsional GW modes, which are nondispersive and make it possible to calculate the location of the echo's source through the time of flight [8].

SwRI was allowed to install the MsS corrosion monitoring system on the shell of a low-pressure feedwater heater at one of the commercial nuclear generating stations.

The system collected monitoring data for 747 days between January 27, 2011, and February 12, 2013. The data were collected once a day from all 17 sensors. The timeline of daily data collection is shown in Figure 4. Data collection was intermittent between April and August 2012 because the system was deactivated for short periods of time for computer repairs. In August 2012, data collection stopped completely, and resumed in December 2012. The sensor layout on the heat exchanger shell is shown in Figure 5, which depicts the cylindrical part of the heat exchanger if it were laid flat on a plate [9]. Three of the 20 planned sensors could not be installed due to issues with insulation removal; the remaining sensors are numbered from 4 to 20. Of the 17 sensors, seven are positioned to direct the wave

circumferentially, while 10 are positioned to direct the wave axially. This was done to improve volumetric coverage of the shell area under investigation. One of the circumferential sensors, No. 17, failed due to demagnetization shortly after installation and was out of service for the duration of data collection [9]. The sensors are indicated in Figure 5 by light green rectangles with numbers. Two large dark brown-green circles in the middle indicate inlet nozzles. Other dark brown-green circles and rectangles represent piping and structural components. For the circumferential sensors, the beam may travel around the shell multiple times, thus producing multiple reflections of the same feature. By feature, it is meant that any structural, geometric, engineering, or degradation discontinuity can produce a reflection. Figure 6 shows different ways the beam can travel around the shell circumferentially. The circumference of the shell is 167.8 in., while the length is 534 in.

Figure 4. Data collection timeline.

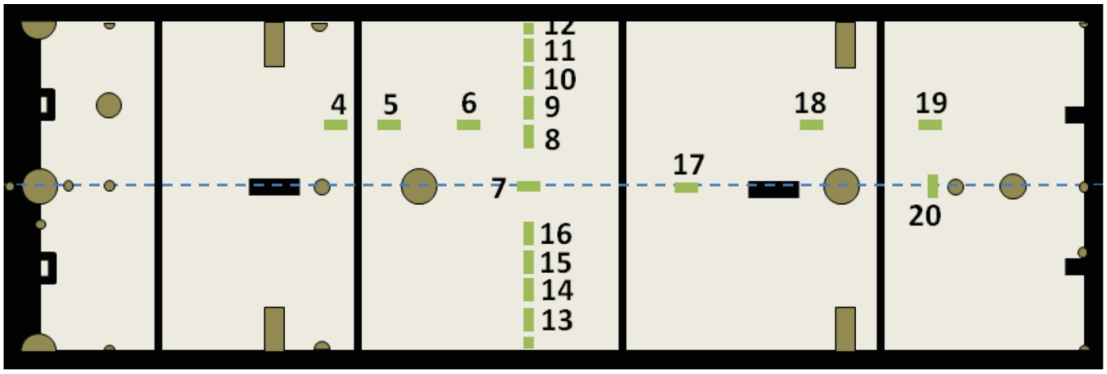

Figure 5. Sensor layout on heat exchanger shell (figure source [8]).

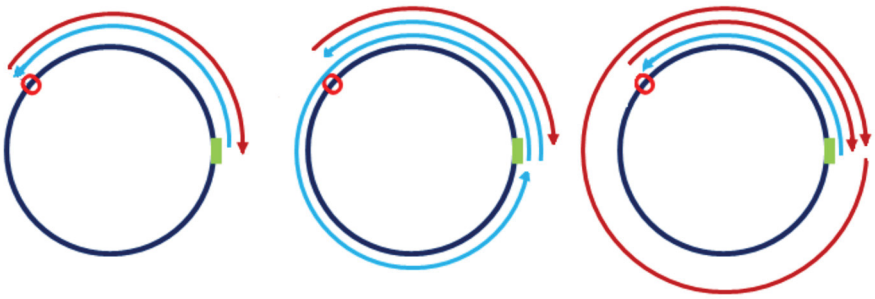

Figure 6. Three different ways a wave can travel circumferentially around the shell propagating in one direction. The sensor is shown as the green rectangle, the feature is the red circle, the pulse is blue, and the echo is red (figure source [8]).

The volumetric beam coverage for circumferentially oriented sensors is shown in Figure 7. It can be seen that the beam coverage for some of those sensors overlap significantly, meaning that the same feature will be registered on adjacent sensors. This means that the data from those sensors can be processed as a group, which is very important while applying ICA because one of the fundamental assumptions is that different sensors receive information from the same source. For this report, only data from circumferential sensors were analyzed.

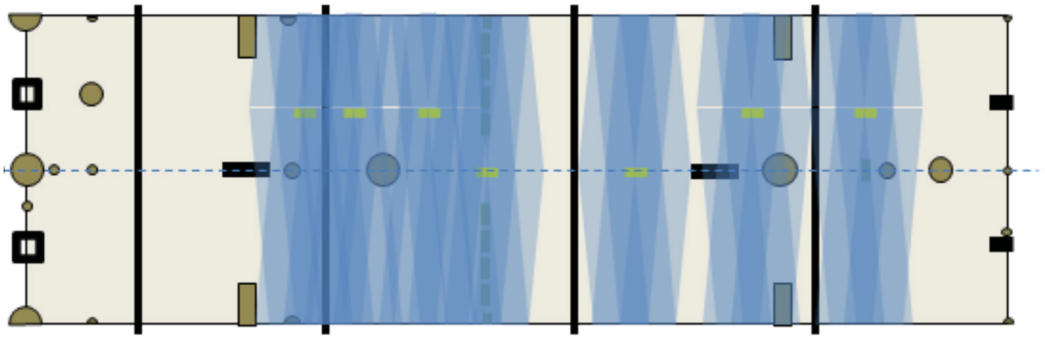

Figure 7. Beam coverage for circumferential sensors (figure source [8]).

2.1 Independent Component Analysis and Blind Source Separation

For UGW to be a competitive technology in the nuclear industry, it needs to overcome several shortcomings, such as low sensitivity to minor degradation, dependence on geometry, and a low signal-to-noise ratio (SNR) in a heavily degraded environment. This report aims to address the low SNR by applying advanced signal processing and pattern recognition techniques capable of dealing with coherent noise, which is endemic in UGW systems, and low SNR.

The echo signals recorded with UGW systems usually consist of several peaks that correspond to reflections from different features of the structure under inspection, such as welds, supports, elbows, or areas of corrosion and erosion. In addition to these peaks, there is background noise that mainly happens due to the following reasons: (a) the material will usually exhibit low-level surface roughness that is caused by the interaction of the ultrasonic signal with the structure; and (b) ultrasonic mode conversions.

When the ultrasound interacts with a feature, coating, or surface roughness, some of the energy will be converted into different wave modes. If a mode is dispersive, it will contribute to the background signal (noise) as it spreads out in time and space. This is sometime called shadowing.

Ideally, a UGW technique will only transmit the non-dispersive wave modes; however, interaction of the ultrasonic wave modes with non-axisymmetric features of the structure can lead to mode conversions. This results in the generation of dispersive wave modes (DWMs). To increase the defect sensitivity and improve the SNR of reflection from features, it is important to filter out the DWM as much as possible.

The background noise produced by dispersion is coherent (non-random) and overlaps in the frequency domain with the signal of interest. Since the DWM is non-random, it cannot be tackled through averaging or other noise-reducing techniques. Conventional filtering techniques, such as low-pass and high-pass filtering or averaging, are also unable to reduce this non-random narrow-band background noise.

ICA is a technique that can deal with non-random coherent noise. This technique uses knowledge of the dispersion characteristics of the wave mode, such as non-Gaussianity, and deconvolves signals in the time domain. It does not rely on frequency characteristics, but rather on statistical properties of signal and noise. This technique is an extension of a traditional statistical method—principal component analysis (PCA) [10]—and extends it to the independent signals, while traditional PCA can only tackle uncorrelated signals. In the case of Gaussian random variables, independence and lack of correlation are equivalent, which is why the assumption of non-Gaussianity becomes critical for this technique. However, in the case of UGW testing, this assumption is perfectly legitimate since the reflected waves are periodic signals with non-Gaussian distributions.

The basic PCA approach linearly transforms a data matrix of n columns (sensors) and p rows (observations) to an orthogonal principal components space of equivalent dimensions [10]. The transformation occurs such that the direction of the first principal component is determined to capture the maximum variation of the original data set. The variance of subsequent principal components is the maximum available in an orthogonal direction to all previously determined principal components. The full set of principal components is an exact copy of the original data set, though the axes have been rotated. Selecting a reduced subset of components results in a reduced dimension structure with the majority of the information available, in which information is assumed to be equivalent to variance. Usually, small variance components that are not retained are assumed to contain unrelated information, such as process or measurement noise. While PCA looks to decorrelate a signal's components, ICA aims to make them statistically independent.

ICA was introduced in the early 1980s and attained wide attention and growing interest in the mid-1990s. The technique attempts to identify original signals from a mixture of observed signals, which are a linear combination of sources, without knowing any information about the mixing matrix, or having any prior information about the sources except that they are assumed to be independent and have non-Gaussian distributions.

Since the sources are assumed to be independent, they are termed independent components (ICs). One requirement for IC isolation is that only one component can have a Gaussian distribution. A detailed survey of ICA can be found in [11,12,13]. ICA is rooted in the need to find a suitable linear representation of a random variable. The classic method to solve this problem is to use second order information in the

covariance matrix, such as PCA and factor analysis [10,14]. Rather than applying independence as a guiding principle, PCA attempts to linearly transform a data set resulting in uncorrelated variables with minimal loss of information [13]. For Gaussian-distributed variables, uncorrelatedness is identical to independence. For non-Gaussian-distributed variables, independence is a much stronger requirement than uncorrelatedness.

For non-Gaussian data, higher-order statistics are needed to obtain a meaningful representation. Projection pursuit is a technique for finding interesting projections of data, such as clusters using higher-order statistics [14]. Projection pursuit uses a cost function, such as differential entropy [14], rather than the mean-squared error used in the PCA transformation. Projection pursuit is effective for non-Gaussian data sets. There are similarities and connections between ICA and these techniques. In the noise-free case, ICA is a special case of projection pursuit. ICA can also be viewed as a non-Gaussian factor analysis. ICA must use higher-order statistics while PCA only uses second-order statistics.

ICA is a statistical framework in which the observed data, X, are expressed as a linear transformation of latent variables ('ICs', S) that are non-Gaussian and mutually independent. We may express the IC model as:

$$X = A \cdot S \tag{1}$$

where X is an (n x p) data matrix of n sensors each containing p observations, S is an (n x p) matrix of p ICs, and A is an (n x n) matrix of unknown constants, called the mixing matrix.

The problem is to determine a constant (weight) matrix, W (n x n), so that the linear transformation of the observed variables Y (n x p)

$$Y = W \cdot X \tag{2}$$

has some suitable properties. In the ICA method, the basic goal in determining the transformation is to find a representation in which the transformed components, y_i, are as statistically independent from each other as possible and hopefully represent S.

When random variables with specific non-Gaussian distributions are combined, the central limit theorem states that the sum is more Gaussian than the original variables. Therefore, to separate the original variables (S) from a sum (X), we want to choose a transformation (W) that makes them as non-Gaussian as possible. We then assume that the maximally non-Gaussian signals, Y, are estimates of the original ICs, one of which is the parameter value, and thus the parameter estimate.

Hyvarinen [11] developed a particularly efficient ICA algorithm called *FastICA*, which is used in this report. It uses negentropy $J(y)$ as the measurement of the non-Gaussianity of the components:

$$J(y) = H(y_{gauss}) - H(y) \tag{3}$$

where *H(y)* is the differential entropy of a random vector *y*, which can be written as:

$$H(y) = -\int f(y)\log(f(y))dy \tag{4}$$

where *f(y)* is the probability density function of random vector *y*.

Based on the maximum entropy principle, negentropy *J(y)* can be estimated as [11]:

$$J(y_i) \approx c[E\{G(y_i)\} - E\{G(v)\}]^2 \tag{5}$$

where *G* is any nonquadratic function, *c* is a positive constant, *v* is a Gaussian variable of zero mean and unit variance, and *E{}* is the operator of mathematical expectation. ICA has two ambiguities. One is that the variances of the ICs cannot be determined. The other is that the order of the ICs cannot be determined [13]. In this report, the *FastICA* algorithm has been utilized to perform ICA on UGW data collected on a steam generator shell.

2.2 Performance of ICA on Collected Guided Wave Signals

Applying UGW technology, it is possible to inspect long stretches of straight pipelines from a single location; a single collar of sensors can routinely inspect up to 100 m of pipe [15] under ideal conditions. However, such long coverage is a compromise between length of inspection and sensitivity to changes in cross-section with most UGW systems capable of resolving up to 5% of wall thickness change [16]. This is the reason for combining UGW with gridded ultrasonic inspections in NPPs [9]. The GW has full volumetric coverage and propagates well in steel, making it especially suited for long-range screening applications without many bent pipes.

When applying ICA to UGW data, it is postulated that the matrix *X* contains reflections from *n* sensors each having *p* samples as a function of time or distance. The matrix *S* is the sought-after representation of *n* ICs each having *p* time samples. Columns of matrix *A* are weighting functions and indicate how much each source signal contributes to each observed signal. Since there are two sets of sensors installed on the heat exchanger shell—circumferential and axial—these two sets need to be analyzed separately as two groups. Also, for ICA to be applicable, the sensor beams need to volumetrically overlap as shown in Figure 7. For this reason, the group of sensors that were analyzed in this report using ICA were four circumferential sensors: S4, S5, S6, and S7. Since the GW propagates around the shell multiple times, the same features are detected repeatedly after each circle. Figure 8 shows reflection signals from the four circumferential sensors, along with major identified features. Four round trips around the shell are shown since some features are only detectable after the second- or third-round trip, after the beam has spread [8,9]. The position of each feature is plotted against the metal path distance

from the sensor. The location of structural, engineering, and geometric features is known from the shell's layout plan provided by the plant.

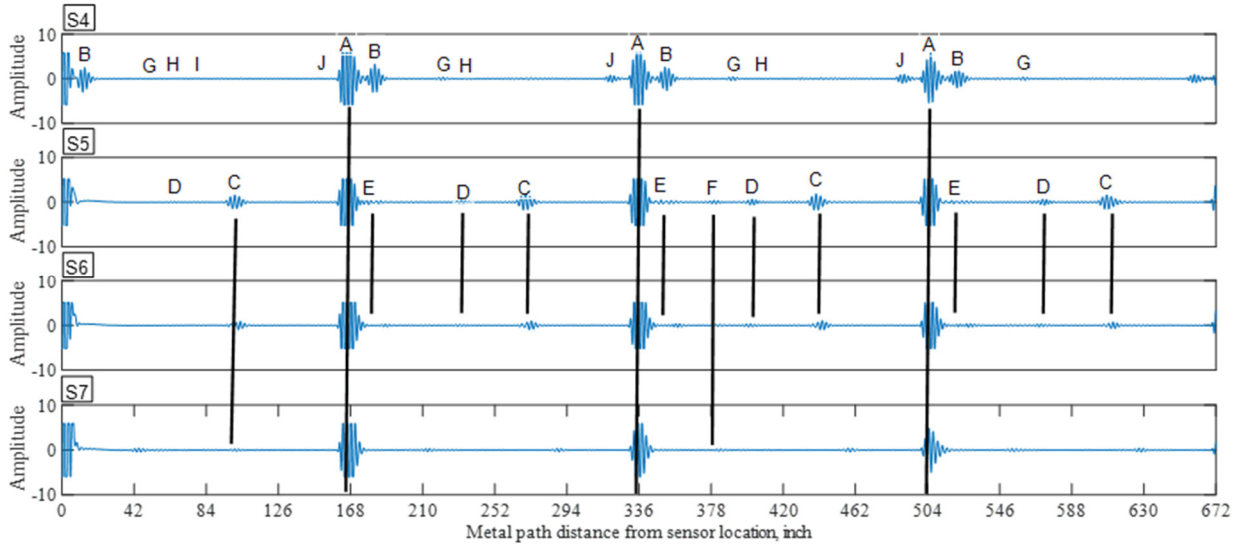

Figure 8. Reflection signals from circumferential sensors S4, S5, S6, and S7. A–sensor-transmitted signal; B–axial weld located on the south side of the shell; C–axial weld located on the north side of the shell; D–unknown source; E–unknown feature, only seen after the second-round trip due to the beam spread; F–reflection from the south side of the inlet nozzle; G–reflection from the south side of the small pipe at the top of the shell; H–reflection from north side of small pipe at the top of the shell; I–unknown source; and J–unknown source.

Since each reflected signal is a mixture of different sources, the goal of ICA is to separate the sources into different components, thus extracting defect or structural reflections into one set of components and relegating noise to other components. Due to multiple round trips around the heat exchanger shell, the reflected signal from one sensor is almost a complete replica repeated four times. The first segment runs from 0 to 167.8 in., the second from 167.8 to 335.6 in., the third from 335.6 to 503.4 in., and the fourth from 503.4 to 672 in. Analyzing all four replicas or segments does not add any new information; as such, the ICA was focused on smaller segments of the reflected signals. Notice, not all features were detected by all sensors. This is because not all structural and geometrical features were in the signal's path from a particular sensor and because reflections from some features detected by a specific sensor were too weak to be visible on the graph.

The first analysis was performed on segment one to see if ICA can separate the pulse signal A into a separate component. The pulse signal A after a round trip is registered by all four sensors and is located at a distance of 167.8 in., which is the circumference of the shell. It is replicated at 335.6 and 503.4 in., as shown in Figure 6. The feature A is essentially the sensor registering its own pulse after the pulse made a round trip around the shell.

Figure 9 shows ICA applied to the first echo signal A as shown in Figure 8. The right-hand panel in Figure 9 shows the raw echo signals registered by four circumferential sensors. As we can see, all four sensors pick up the pulse signal after its first round trip around the shell. The left-hand column in Figure 7 shows the ICs extracted from four raw signals. As we can see, IC 2 contains the echo signal, while the other three components contain noise. While the separation is not perfect, this result demonstrates that ICA is capable of separating the echo component from the noise. While IC 2 has some distortions in comparison to the raw signal, it does contain all elements of the echo signal. On the other hand, the other three ICs represent noise, which has been extracted from the raw signals. This example has one clearly defined source and four sensors registering it. The next example is more challenging as we process the data containing reflections from multiple segment three sources in Figure 8. This segment contains reflection B from the weld, the unknown source reflection E, and some noise. The signals for this analysis were taken from the third-round trip segment, since feature E seemed to have a better reflection after the third-round trip due to beam divergence. The results are shown in Figure 10. As we can see, the weld reflection component and the unknown source reflection are separated into ICs 1 and 3, while the noise components are relegated to ICs 2 and 4. IC 2 contains broad-band noise, while IC 4 contains narrow- and broad-band noise. Notice, that after ICA, IC 3, which mostly contains feature E, has an amplitude comparable with IC 1, which is the weld reflection. In the raw data, the weld reflection B has an amplitude that is four times larger than feature E. In this case, ICA is helpful in improving the detection of weak features.

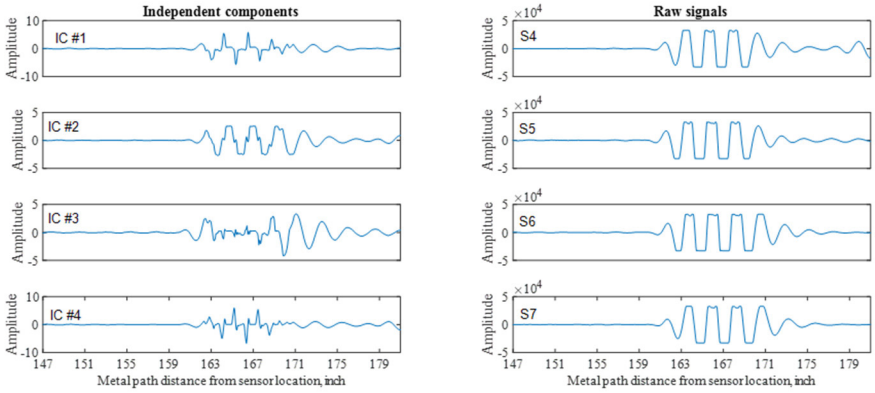

Figure 9. Results of applying ICA to sensor-transmitted signal A.

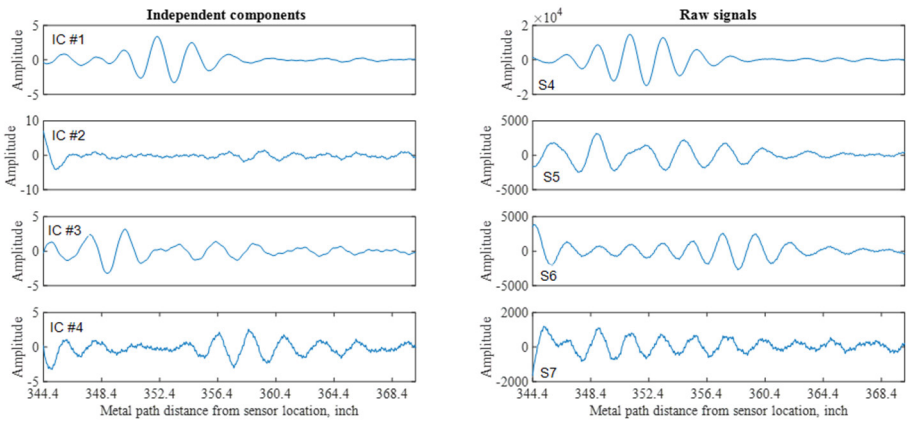

Figure 10. Results of applying ICA to the weld reflection C.

Finally, ICA is applied again to the first segment in Figure 8 that contains multiple reflections, such as the C-axial weld located on the north side of the shell, reflection D from an unknown source, reflection G from the south side of the small pipe at the top of the shell, reflection H from the north side of the small pipe at the top of the shell, and reflection I from an unknown source. The results of this analysis is shown in Figure 11. In this case, ICA is challenged to work with a situation when the potential number of sources—five—is larger than the number of sensors—four. In industrial applications, such a situation may arise when trying to detect pitting corrosion that will have numerous reflections from small defects and the number of defects may well exceed the number of sensors. Figure 11 shows that in this case, the ICA does not perform well as it is unable to extract specific features into different components. Although this result is expected due to limitations of the currently used algorithms, it points to the need to develop ICA algorithms that can tackle situations with a large number of sources, potentially exceeding the number of sensors. These new approaches need to focus on providing additional information about fixed and unchanging reflection sources, such as welds and nozzles. Supplying such information to algorithms will allow analysis of new and unknown sources that may be due to the corrosion process.

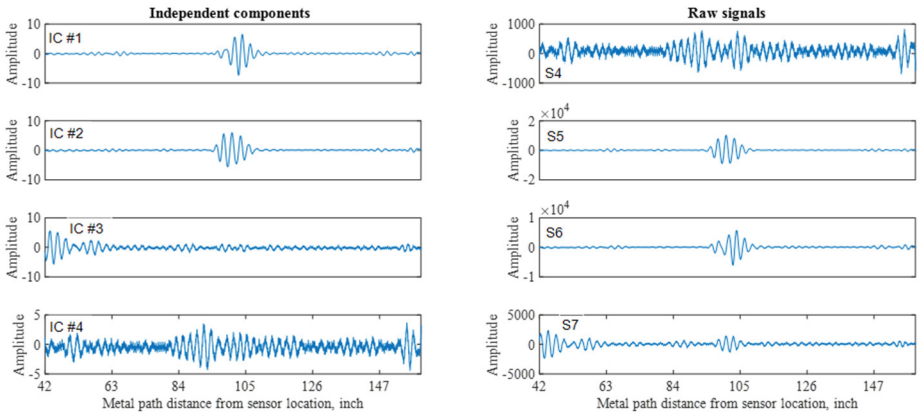

Figure 11. Results of applying ICA to the first segment of the data with multiple reflections in Figure 6.

3. ADVANCED ML PATTERN RECOGNITION TECHNIQUES TO PROCESS DATA FROM UGWS AND FIBER OPTIC TRANSDUCERS

3.1 Shallow BackPropagation Neural Networks and Regularization

The ICA is a feature extraction algorithm producing relevant features that can be further utilized for pattern recognition using advanced ML algorithms, such as neural networks (NNs) or support vector machines (SVMs). Both techniques are widely used for regression estimation and pattern recognition. Since they are nonlinear and nonparametric, they offer unique capabilities for both regression and pattern recognition; however, due to their nonlinear and nonparametric nature, they can also cause problems with the development of N models.

One of the most challenging problems with NNs is their regularization, which is the ability to obtain repeatable and consistent results regardless of small variations in training data or the network's architecture. NNs require nonlinear regularization, which is a much more difficult problem than its linear counterpart, and has no general solution due to nonlinear error propagation and existence of multiple local minima (multiple solutions) on the error surface. Once the NN training is complete, the problem then becomes the estimation of its generalization capabilities or, in other words, has the gradient decent converged to a correct solution or is it necessary to change the NN architecture or initialization to look for a better solution. The solutions provided by the NNs depend on several factors, namely, weight initialization, number of neurons, stopping criteria, and the training algorithm. To make pattern recognition performed by NN consistent, it is necessary to make the NN's solution invariant under all of these different conditions. If we are not able to get consistency under these different conditions, it will become necessary to at least estimate the reliability of our inference.

Several methods have been proposed to assure the stability and consistency of the NN's solutions. In this report, we tested the most popular NN's regularization techniques (i.e., the Levenberg-Marquardt [LM] algorithm, weight decay, cross-validation, weight initialization, and Bayesian regularization). The NN under investigation here involved multilayer perceptron (MLP) with nonlinear hidden layer and step output function. There were 513 input neurons, 10 hidden neurons, and 2 output neurons. This network's architecture is shown in Figure 12. The input patterns have been obtained by applying Fourier transform to the UGW signals, and then converting Fourier transform into power spectral density (PSD). The UGW signals and their power spectral densities are shown in Figure 13.

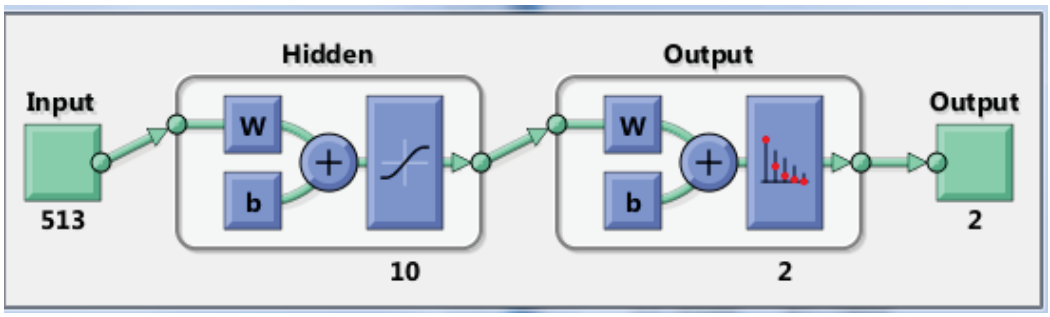

Figure 12. NN architecture used for the pattern recognition of UGW signals.

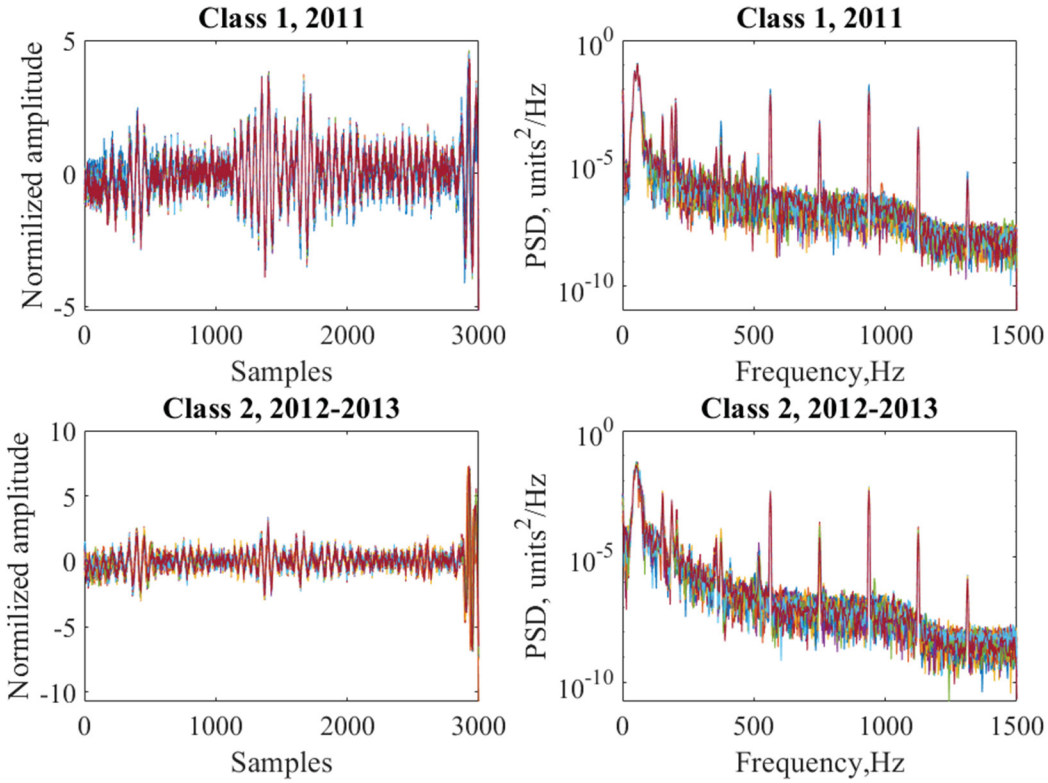

Figure 13. UGW signals from sensor 4 used to train, validate, and test the NN. The left-hand column shows the time-domain signals for the two classes, while the right-hand side column provide the PSDs for the two classes.

The UGW signals used for MLP training were selected from circumferential sensor #4. Class 1 signals were recorded between January 27 and March 16, 2011, while Class 2 signals were recorded between December 26, 2012, and February 12, 2013. For each class, 49 patterns have been created, each pattern representing the PSD of a signal recorded on one day. The last pattern for Class 1 recorded on March 16, 2011, and the first pattern of Class 2, recorded on December 26, 2012, are separated by nearly 19 months, which is sufficient time for the development of a noticeable degradation in the heat exchanger shell. To have a reference for NN consistency, the MLP was first trained without any regularization down to mean square error 10^{-3} using ordinary gradient decent. Cross-validation was used to prevent overfitting.

There were 10 hidden neurons and their activation function was via hyperbolic tangent. The training was performed one hundred times starting from different initial weights with a limiting number of epochs equal to 1000. The classification accuracy had been estimated using a test data set. For training, the available 98 patterns for two classes were randomly divided into a training set, a validation set, and a test set. The classification accuracy was calculated for the whole data set representing all 98 patterns.

The LM algorithm has inherent regularization properties, as discussed in [17]. To study the regularization properties of the LM algorithm, an NN was trained one hundred times starting from different initial conditions and for a different number of training patterns. The results of this exercise are shown in Table 1.

Table 1. Average classification accuracy and its standard deviation for different methods of NN regularization. The NN had 513 input neurons, 10 hidden neurons, and 2 output neurons.

Training Method	Average classification accuracy, %	Standard Deviation of classification accuracy, %
Gradient descent	74.4	23.5
Levenberg-Marquardt	99.9	0.8
Weight decay regularization	85.1	20.5
Bayesian regularization	96.7	10.3
Cross-validation regularization	98.5	8.5
Regularization through weight initialization	87.3	19.2

In Table 1, “average classification accuracy” denotes the mean value of classification accuracy calculated for 100 runs with different initial weights and randomly selected training patterns. The standard deviation reflects the variability in classification accuracy from run to run. As can be seen from Table 1, the average value of classification accuracy for gradient descent depends significantly on weight initialization and training patterns with a mean value of 74.7% and a standard deviation over 23%. The standard deviation shows that NN inference is unstable under different random starts when other parameters, such as the number of hidden neurons and the training method are fixed. However, it should be noted that for the LM method of regularization, the variance of classification accuracy was substantially reduced when compared to ordinary unregularized gradient decent. Using the LM algorithm improved the classification accuracy to over 99%, which is a significant improvement over ordinary gradient decent. In addition, the standard deviation of classification for the LM method is dramatically reduced, as is evident in Table 1.

The instability of inference for gradient descent can be attributed to redundant flexibility of NN as a function approximator and to collinearity of the training data set. It has been known for a long time in the NN community [18,23,24] that to get a network with good generalization capabilities, some kind of capacity control should be imposed on the family of functions that can be implemented by a neural net. For example, this type of control can be implemented by controlling the magnitude of weights and biases

in the neural net. As shown in [19,20], the complexity of the function that can be implemented by the NN depends on the magnitude of the weights (i.e., the bigger the weight, the more complex function the NN can approximate). Obviously, a sufficiently large NN with a large number of hidden neurons can approximate the arbitrary complex function up to any degree of accuracy. The problem is that letting the NN do this also allows it to approximate noise or artifacts in the data, thus "discovering" "structures" that do not actually exist in the data; hence, providing a predictive model with poor generalization performance on new unseen data.

By constraining the NN complexity, it is hoped that a subtle compromise between fitting the data and keeping our model as simple as possible can be resolved. This is a version of Occam's razor, which states that a simple model should be preferred to a complex one provided both are consistent with the data. The easiest way to restrict the complexity of an NN is to add a penalty term to its least square error function. This penalty term is usually the sum of all the squares of the NN's weights and biases and is analogous to ridge regression in statistics. The penalized functional to minimize in this case looks like:

$$\text{Total Performance} = \lambda E + (1-\lambda) S \tag{6}$$

where E is the usual mean squares error term, S is the penalty term, which is the squared norm of all weights and biases in the network, and λ is the regularization parameter that controls the trade-off between E and S. The rationale behind this type of regularization is that we anticipate that mapping, implemented by NN, should be smooth or non-oscillating, and that the second term in formula (6) penalizes such non-smoothness. The parameter λ should be chosen prior to the application of this method of regularization. The selection of regularization parameter is a difficult problem even for linear techniques and will be addressed later in this report. Parameter λ is defined by the amount of noise in the data, which is usually not known *a priori*. Some initial ideas about the value of this parameter can be delivered by analysis of the eigenvalues of the Hessian matrix for the NN, as is shown in [21].

The results of NN training with weight decay regularization is shown in Table 1 for $\lambda = 0.9$, which is selected by cross-validation. The network was reinitialized 100 times, every time with the same λ to check the dependence on the random start. This dependence is summarized by standard deviation of the classification accuracy in Table 1. As can be seen, weight-decay regularization improved NN's stability, while at the same time significantly improved its classification accuracy. However, it has not been able to match consistency or accuracy of the LM method. Under certain assumptions, the weight decay regularization has a probabilistic Bayesian interpretation. Consider this learning problem from a general point of view as an estimation of an unknown function from the available finite amount of data [23]:

$$y_i = f(x_i, w) + \varepsilon_i \ , \qquad i = 1 \ldots N \tag{7}$$

where y is response variable, x is the vector of independent variables, w is vector of model parameters that have to be estimated during the training process, ε is the noise term that is assumed to have normal distribution with zero mean and some variance σ^2, and N is the number of available data samples. According to Bayes theorem, *a posteriori* probability of the model having the data P(w/[y, x]) is proportional to the likelihood of data assuming that the model is true multiplied by prior probability P(w) of the model, or:

$$P(w/[y, x]) \propto P([y, x]/w)*P(w) \tag{8}$$

where P ([y, x]/w) is the likelihood of observing the data given a particular set of weights in the model. This likelihood is just a joint probability density function for observed data that has the set of weights *w* as a parameter. Assuming independence of different training points and taking into account the normality assumption, we get:

$$P([y, x]/x) = \prod_{i=1}^{N} Gauss(y_i - f(x_i w)) \tag{9}$$

Making additional assumption that prior distribution of weights is Gaussian with zero mean and some variance and taking logarithm of both parts in (9), we get a familiar penalized functional:

$$\ln(P(w/[x,y])) \propto \frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i, w))^2 + \frac{2\sigma^2}{N}\|w\| \tag{10}$$

This is a maximum *a posteriori* estimate or penalized maximum likelihood estimation with explicit form of regularization parameter, which is obviously a function of unknown noise variance σ^2. This analysis demonstrates that simple weight decay under the assumption of normality of the error term is equivalent to one of the forms of Bayesian inference with Gaussian prior on weight distribution.

The second very popular type of regularization for NNs is early stopping or cross-validation. This kind of regularization is largely ad hoc and is based on dividing the training data into two parts: the training set and the validation set. The idea is to stop ML before the NN begins to learn noise and spurious structures in the data. During the training, while minimizing mean squared error, the NN learns more and more structure from the data; however, at some point the NN begins to learn a pseudo-structure or noise, thus providing more "rough" mapping. The goal of the validation set is to provide an independent test set for verification of how well the trained network is going to generalize on previously unseen data. The results for this kind of regularization are shown in Table 1. For this test, 30 training patterns were run with 10 validation patterns. It can be seen from Table 1 that the classification accuracy again depends on the number of on-random initialization, as indicated by the standard deviation of 8.5%.

The standard deviation is relatively small in comparison to the gradient descent and weight decay regularization, but significantly larger than LM's standard deviation. It should also be noted that the average classification accuracy of 98.5% is high and second only to the LM method. An obvious limitation of this type of regularization is that the final solution depends on the initial start, as well as the path by which the system evolved to its final state. In addition, it requires splitting training data into at least two sets, thus decreasing the amount of data available for training, which in case of scarce data can be a serious limitation. An obvious advantage of this kind of complexity control is its simplicity.

A less-known type of regularization for NNs is regularization by initialization, when initial weights of the NN are set to small values, thus forcing the NN to converge to the same local minima and minimizing the dependence on the random start. This kind of regularization is also ad hoc because by setting the initial weights to small values, the solution can, in fact, be specified. The application results of this kind of regularization is shown in Table 1. It can be seen that regularization by initialization reduced the standard deviation in comparison to gradient descent, meanwhile increasing classification accuracy. However, this type of regularization cannot match the accuracy or consistency of the LM method or regularization by cross-validation.

The most advanced method of NN regularization is via Bayesian regularization [19,20,22]. The Bayesian point of view on NN training is rather different from traditional. The traditional methods are variations on the maximum likelihood principle, which states that from a variety of possible models the one that should be picked up is the one most probable to the observed data. The maximum likelihood principle considers model parameters as unknown but fixed values, and tries to estimate these parameters from the available data, providing the only set of parameters most likely generated by the observed data. In conventional NN training, a single set of weights are available, which are used for future inference. In contrast to the maximum likelihood principle, the Bayesian approach considers the model parameters to be random variables having *a priori* distribution. Having obtained this prior distribution, the Bayesian inference proceeds with an application of the Bayes theorem to modify this prior distribution and produce *a posteriori* distribution that now depends on prior information and the data. As such, the Bayes theorem for statistical learning can be written like this:

$$P(Model / Data) = \frac{P(Data / Model) * P(Model)}{P(Data)} \tag{11}$$

where the P(Model/Data) describes the conditional probability that a Model is true given Data. P(Data/Model) describes conditional probability to observe Data given a Model, or in other words, the likelihood of observing the data if the Model is true. P(Model) describe our prior beliefs, in the form of

probability, how an actual model is expected to look. P(Data) is the total probability to observe the Data under all thinkable models described by P(Model). The denominator P(Data) does not depend on the model and is sometimes omitted from the formula. The Bayesian approach also allows the comparison of several potential models based on their "evidence," as derived from the data [19]. Bayesian learning for NNs consist of several inference levels [20]. First, we specify the performance function to be optimized in the form:

$$M(w)=\beta ED+\alpha EW \tag{12}$$

where E_D is the data dependent term, E_W is stabilizing term, and α, β are regularization parameters to be defined from the data. At the first level of inference, Bayesian training infers weight values for given regularization parameters α and β, using the Bayes theorem as follows:

$$P(w/D,\alpha,\beta,H)=\frac{P(D/w,\alpha,\beta,H)*P(w/\alpha,\beta,H)}{P(D/\alpha,\beta,H)} \tag{13}$$

where P(w/D,α,β, H) is posterior weights distribution, P(D/w, α,β,H) is the data likelihood given weights w, the regularization parameters α and β, and model H, P(w/ α,β,H) is the prior weights distribution and P(D/ α,β ,H) is the total data probability. Parameters α and β are called hyperparameters to distinguish them from the true parameters—weights. The second level of inference is to infer these hyperparameters, again using the Bayesian approach:

$$P(\alpha,\beta/D,H)=\frac{P(D/\alpha,\beta,H)*P(\alpha,\beta,H)}{P(D/H)} \tag{14}$$

where all of the probabilities in the formula have the usual Bayesian interpretation. The final step in the Bayesian inference is model inference or model selection, when different models are compared based on their "evidence" P(D/H), which is the likelihood for the model multiplied by the model's Occam factor [24], which is the term used to penalize the model for having an excessive number of parameters.

The key to successfully using Bayesian training is the right choice of prior distribution and is sometimes considered to be a shortcoming of Bayesian inference, due to its "subjective" nature. However, facing ill-posed problems, there is no other way to do so but by using prior information, because the data underdetermines the solution. Having chosen prior distribution of the weights, Bayesian training gives rise to posterior weights distribution, which in its turn gives rise to the distribution of the output values during the inference on the new data. The mean of this output distribution is the inferred value.

The application results of using Bayesian regularization to UGW signals classification are shown in Table 1. The number of training patterns used in this experiment was 30. As can be seen from the table,

Bayesian regularization produced a high classification accuracy of 96.7% while significantly reducing the standard deviation in comparison with the gradient descent method. However, it was unable to match the classification accuracy or consistency of LM or cross-validation. Also, this kind of inference drastically depends on the initial number of hidden neurons.

In conclusion, we can say that NNs being a powerful and flexible tool for non-parametric modelling and inference are a tough challenge from the point of view of their regularization and consistency due to inherent nonlinearity. Being unregularized, NNs can provide inconsistent results, which are non-interpretable and non-repeatable.

Our results show that the ML method provides the best solution in terms of classification accuracy and stability. Bayesian training provided a good classification rate and stability in comparison with gradient descent training. However, a very serious limitation of using the Bayesian approach to NN training is its computational burden, which in fact limits its application to a small amount of data and small networks. The use of this approach in online systems is out of the question due to the same reason. For a quick analysis of classification system stability based on NN, cross-validation, and weight decay methods can be recommended, which provide a reasonable trade-off between stability and computational time. The LM algorithm proved to be the most stable technique. The stability of this algorithm can be explained by its built-in regularization properties, which helps it to damp high-frequency noise in weights in the vicinity of the solution. Regularization by initialization is a rather new technique and its validity has to be evaluated more rigorously in theoretical and practical aspects, but our results show that it can reduce its dependence on initial conditions, which is natural to expect, and this method can be effective from a computational point of view because it does not require any additional computational efforts. To obtain results reported in this section, the NN architecture, such as the number of layers and neurons in each layer were kept fixed. If these parameters were varied, then even for most stable methods such as ML, it is expected to have more variability in the classification rate.

3.2 Support Vector Machines

An SVM is an advanced pattern recognition and regression technique that attempts to address existing problems with the ill-posed nature of NN training. Results presented in the previous section demonstrated that NNs are poorly controlled learning machines, which performance depends on initial conditions and training patterns.

The goal of SVM training is to minimize the expected prediction risk on future data. In its most general setting, the problem of learning from the data can be formulated as follows: given a data generating mechanism, which is represented by a joint distribution $P(\mathbf{x}, y) = P(\mathbf{x})P(y/\mathbf{x})$, where $\mathbf{x} \in R^N$

and y ∈ R or {-1,1} depending on what problem we consider—regression or pattern recognition. P(**x**) is a probability density function of input patterns and P(y/**x**) is the probability of output conditioned on the input. In practice, instead of having probability distribution P(**x**, y), we always have samples from this distribution $D_n=(\mathbf{x}_1,y_1),\ldots(\mathbf{x}_n,y_n)$. Given a set of parameterized functions $f_w(x)$, the goal is to find parameters w that minimize the discrepancy between y and $f_w(x)$. This discrepancy is called a loss function and can be written as L (y, $f_w(x)$). A typical example of a loss function is L_2 loss function, which is $L_2 = (y-f_w(x))^2$ or the squared distance between target y and predicted value $f_w(x)$. The expected or true prediction risk is defined as a mathematical expectation of L (y, $f_w(x)$) as:

$$R(w) = \int_{X,Y} L(y, f_w(x))P(x, y)dxdy \tag{15}$$

Obviously, the true risk cannot be found in practice because P(x, y) or data-generating distribution is generally unknown, and what is available is only a sample of it—Dn. As a result, the minimization of true risk (1) in practice is replaced by the minimization of empirical risk, which is defined as:

$$R_{emp}(w) = \frac{1}{n}\sum_{i=1}^{n} V(y_i, f_w(x)) \tag{16}$$

Equations (15) and (16) are also often called risk functionals. The goal of a learning algorithm or learning machine is to find a function $f_w(x)$, which delivers the minimum to risk functionals (15) and (16). The major question that statistical leaning theory answers is if a function $f_w(x)$ delivers a minimum to the empirical risk (16), does it deliver the minimum to the true risk (15)? If it does, under what conditions? The law of large numbers states that if the size of training sample D_n is made infinitely large, then R_{emp} would converge to R; however, for a finite amount of training data, R_{emp} can always be made zero by choosing a model from a class of sufficiently complex models. It means that R_{emp} in practice is overly optimistic in providing a biased estimation of the true risk, R. To overcome this difficulty, statistical learning theory [25,26] considers the worst-case scenario or:

$$\sup_w \left| R(w) - R_{emp}(w) \right| \tag{17}$$

which is, it analysis the maximum discrepancy that can occur between true and empirical risk. Under the probably approximately correct (PAC) model, Vapnik [25,26] showed that the bound for the true risk (for the classification problem) holds with the probability 1-η (0≤η≤1) is:

$$\mathrm{R(w)} \le \mathrm{Remp(w)} + \sqrt{\frac{h(\ln(2n/h)+1) - \ln(h/4)}{n}} \tag{18}$$

It can be seen that the expected risk is bounded from above by the sum of two terms—empirical risk and VC confidence. VC confidence represents a complexity penalizing term and depends on the number of training samples n and VC-dimension h. The VC-dimension is the measure of potential flexibility or capacity of the learning machine. For linear systems, the VC-dimension is equal to the number of parameters used in the learning machine. The problem with the VC-dimension is that theoretical estimates of it are available only for simple learning machines. VC-dimension is a purely combinatorial concept that has no connection with topology. Loosely speaking, the VC-dimension of a learning machine measures how many training examples the machine can learn or memorize with zero empirical risk. Obviously, for a machine to be useful and able to generalize, the VC-dimension should be significantly smaller than the number of training samples to avoid memorization. Thus, the VC-dimension provides an estimation of the minimum number of training samples that are necessary to build a model with valid generalization properties. If we are able to estimate VC-dimension of a learning machine, then we can use this estimation in (18) to obtain the upper bounds on the prediction risk. Unfortunately, the estimation of the VC-dimension for non-linear predictive models like NNs is very difficult and only rather loose bounds are available [25,26].

To overcome the problem with the estimation of VC-dimension, Vapnik [25] proposed a new learning paradigm that is able to control the VC-dimension, and hence, the generalization capabilities of a learning machine. We start consideration of SVM with the simplest linear case and then move towards nonlinear cases. Let us have a set of data, which is linear separable as illustrated by Figure 14.

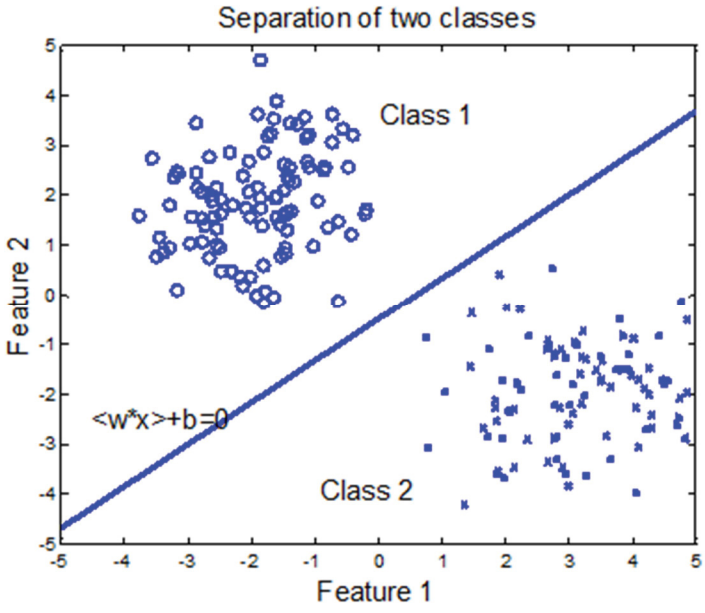

Figure 14. Linear separable classes.

The data are clearly separable, and the optimum separation would be provided by the Bayesian optimal classifier, which however again requires the knowledge of underlying densities. In the absence of information about data distribution, several ways exist to build a separating hyperplane, as is shown in Figure 15.

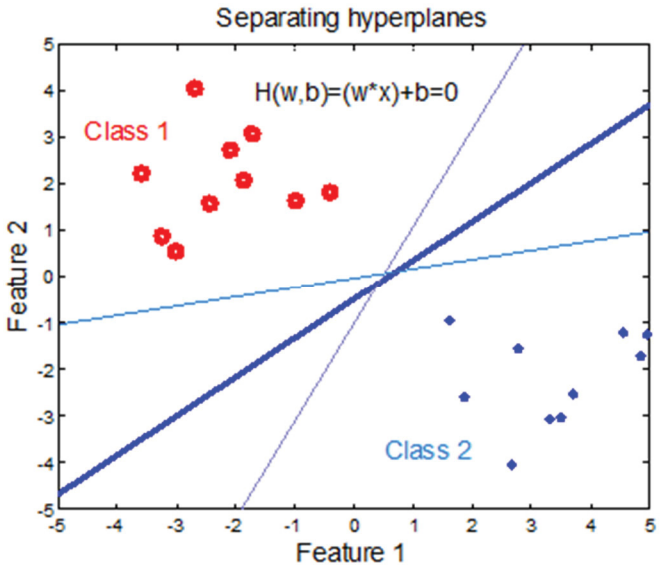

Figure 15. Separating hyperplanes.

Obviously, not all of them would have equal generalization capabilities. Intuitively, the bold line will probably provide the smallest number of errors on future unseen data, but the question remains of how to build it. Vapnik's idea of an optimal separating hyperplane is that the one with the smallest number of future errors should be equally distant from the closest data points. This distance is referred to as the margin and the separating hyperplane is called optimal, if the margin is the maximum size. This idea is illustrated in Figure 16.

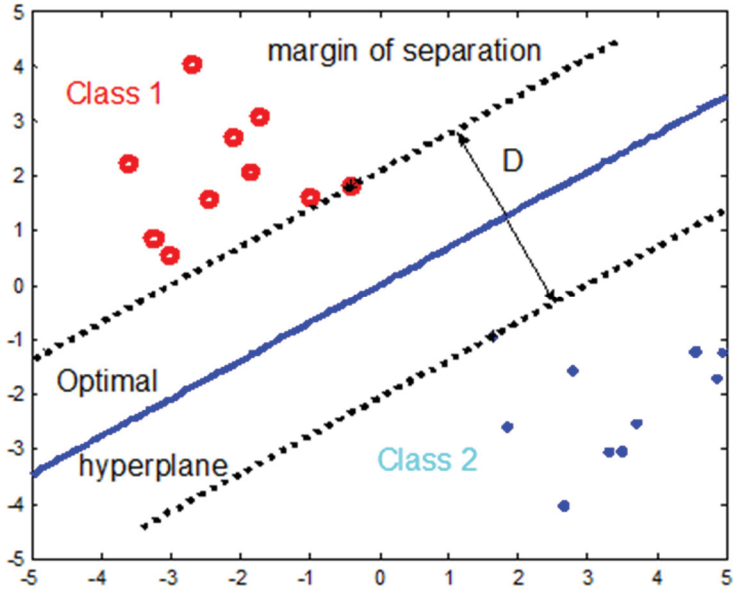

Figure 16. Optimal separating hyperplane.

In our linear separable case, a pair (w,b) can be found such that [25]:

$$w \cdot xi + b \geq 1 \quad \text{for } \forall xi \in \text{Class 1}$$
$$w \cdot xi + b \leq -1 \quad \text{for } \forall xi \in \text{Class 2} \tag{19}$$

and the parameterized hypotheses space in this case would be:

$$f_{w,b} = \text{sign}(w \cdot x + b) \tag{20}$$

Without the loss of generality, we can scale w and b in such a way to get:

$$\min |w \cdot x_i + b| = 1 \tag{21}$$

This normalized hyperplane is called a canonical hyperplane and as shown in [26], the VC-dimension of the canonical hyperplane is N+1 where N is the dimensionality of the input vector **x.** Now we can constrain the set of hyperplanes even more considering only those for which $\|w\| \leq A$. As shown in [26], the VC-dimension of the set of canonical hyperplanes constrained in this way is:

$$h < \min([S^2A^2], n) + 1 \tag{22}$$

where S is the radius of the smallest sphere that contains the training input vectors ($\mathbf{x}_1$,$\mathbf{x}_n$). Hence, controlling the norm of the weights of the separating hyperplane we can, in fact, control the VC-dimension and its capabilities. On the other hand, the distance between a training vector **x** and canonical separating hyperplane is:

$$D(\mathbf{x};\mathbf{w},b) = \frac{|w^* x + b|}{\|w\|} \tag{23}$$

According to the normalization condition (21), the distance between the closest data points and the separating hyperplane is simply $\frac{1}{\|w\|}$, and hence, minimization of the norm of w would lead to the maximization of the distance between the canonical separating hyperplane and its closest data points. Putting all of these together, we need to minimize $\frac{1}{2}\|w\|^2$ subject to the constraints of (19) that can be written in compact form as: $y_I(w^*x_i + b) \geq 1$, i=1,…,n. We use the technique of Lagrange multipliers to construct the Lagrangian:

$$L(w,b,\lambda) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \lambda_i [y_i (w^* x_i + b) - 1], \tag{24}$$

where λ is the vector of non-negative Lagrange multipliers. The solution to this optimization problem is determined by the saddle point of this Lagrangian, which has to be minimized with respect to w and b and maximized with respect to λ [25]. Differentiating (24) and setting the results to zero gives:

$$\frac{\partial L(w,b,\lambda)}{\partial w} = w - \sum_{i=1}^{n} \lambda_i y_i x_i = 0 \qquad and \qquad \frac{\partial L(w,b,\lambda)}{\partial b} = \sum_{i=1}^{n} \lambda_i y_i = 0 \tag{25}$$

From (25), the optimal weights can be written as:

$$w^* = \sum_{i=1}^{n} \lambda_i^* y_i x_i \tag{26}$$

The vectors for which $\lambda_i \geq 0$ are called Support Vectors, and only these Support Vectors contribute to the construction of the optimal solution.

The bias term can be calculated as:

$$b^* = y_i - w \cdot x_i \tag{27}$$

for any support vector x_i. From (20), (25), and (26), the optimal decision surface can then be written as:

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{n} y_i \lambda_i^* (x^* x_i) + b^*\right) \tag{28}$$

Notice that data x and x_i appears in the solution only in the form of a dot product. This plays a vital role in the generalization of SVM for a nonlinear case.

3.3 Nonlinear Support Vector Machines

Linear problems, while helpful in the clarification of ideas, are of little practical use. To generalize SVM to a nonlinear case, Vapnik [25,26] considered the mapping of the input vector **x** into a high-dimensional feature space φ (**x**) with the dimensionality of M>N. Then, the optimal hyperplane in feature space would be:

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{n} y_i \lambda_i^* \varphi(x) * \varphi(x_i) + b^*\right) \tag{29}$$

which is again a hyperplane, but in feature space. The transformation operator φ might be computationally expensive and very difficult to find. However, if there were a "kernel function" $K(x,x_i) = \varphi(x)*\varphi(x_i)$, we would not need the explicit form of the transformation, just this kernel function. All of the previous derivations hold since we are still doing linear separation, but now doing so in the feature space. By using the "kernel function," we can write the separating hyperplane in feature space as:

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{n} y_i \lambda_i^* K(x, x_i) + b^*\right) \tag{30}$$

These "kernel functions" can be constructed considering general forms of the dot product in a Hilbert space [26]. Examples of such functions are:

Simple dot product: $K(x,x_i) = \langle x \cdot x_i \rangle$

Vovk's polynomial: $K(x,x_i) = (x \cdot x_i + 1)^p$

Radial basis function: $K(x,x_i) = e^{-\|x-x_i\|^2 / 2\sigma^2}$

Tree layer neural network: $K(x,x_i) = \tanh(kx \cdot x_i - \delta)$ (31)

Each function implements different types of learning machines – linear, polynomial, radial basis function NN, or perception with one hidden layer. Hence, SVM accommodates different learning machines under one theoretical and implementation roof. The ability of SVM to find the most important patterns in multidimentional space can be used to discover important features otherwise not accessible by other techniques. For this report, we used an SVM with a radial basis function (RBF) kernel. The SVM has been trained and tested on the same 98 patterns used to test NNs in the previous section. The results of SVM performance are summarized in Table 2. For comparison purposes, this table also contains the classification results obtained with the NNs.

Table 2. Performance of SVM classifier in comparison with NN classifiers.

Classification Method	Average classification accuracy, %	Standard Deviation of classification accuracy, %
SVM, RBF kernel	100	0
NN with LM training	99.9	0.8
NN with Bayesian regularization	96.7	10.3

The analysis of Table 2 reveals that SVM demonstrates superior performance in terms of classification accuracy and stability. While the SVM performance is impressive, it may be affected by the set of features that are presented to the leaning algorithm.

3.4 Feature Selection

Prior to utilizing ML pattern classification, different feature-selection methods should be applied to the existing sets of data in order to represent data in compact and most separable format. Feature-selection methods are dimensionality reduction techniques that aim to extract the most relevant information from raw data. Transformation of raw data into a set of features is called feature extraction. Feature extraction is necessary because raw data are often noisy and contain information that may be irrelevant to the problem at hand. In pattern classification, a choice of the "best" feature subset is known as the feature selection problem. In general, the problem of feature selection asks two questions: (1) which feature should be included in the classification model, and (2) in what form should they be included? The answer to these questions depends on a specific application of the classification system. For the defect detection problem, it is necessary to choose a subset of feature variables that have the best predictive power (i.e., minimum prediction error on future data). Unfortunately, it is not straightforward to choose such a subset simply because of the lack of a validation data set that could be used to evaluate the prediction quality.

Refer to a linear model that represents a response variable Y in terms of all available predictor variables $X_1, X_2, \ldots, X_q$ ($q = 24$) as a full model:

$$y_i = \beta_0 + \sum_{j=1}^{q} \beta_j x_{ij} + \varepsilon_i \tag{32}$$

and to a linear model that represents Y in terms of a subset of p variables ($p < q$) as to a subset model.

By fitting the full model (32) to a set of data by using least squares, we obtain the best solution in the sense of minimum Mean Squared Error (MSE) on this set of data. This solution does not guarantee good prediction on future observations. Experience shows that in most applications, prediction accuracy is not improved by simply using all of the available predictors, more often the opposite effect is achieved [27]. In other words, the prediction accuracy of the full model is worse (or at least not better) than those of the

subset models because the variance of the predicted values for linear models with parameters fitted by least squares increases monotonically with the number of variables used in the prediction [28]

$$\operatorname{var}\left(\underset{n\times q}{X} \cdot \beta \right) \geq \operatorname{var}\left(\underset{n\times p}{X} \cdot \beta \right) \tag{33}$$

If a variable has no predictive value, then deleting that variable may increase the precision of the estimates.

However, the price for deleting variables is the introduction of bias to the estimate unless the deleted variables have zero coefficients, or the set of retained variables are orthogonal to the set of deleted variables. Bias means that the value predicted by a subset model is different from that predicted by the full model, assuming that the full model gives an unbiased prediction. However, the solution of the full model is unbiased if the full model is correct and if the noise model is correct as well. In real world problems, neither of these can be claimed to be true. This implies that the solution of the full model may also be biased.

On average, the effect of reduced variance is the deviation of the solution with lower variance $\hat{y}_L$ from the true value y_T (i.e., unknown for any real-world problem) is less than that of the solution with greater variance $\hat{y}_H$:

$$E(\hat{y}_L - y_T) \leq E(\hat{y}_H - y_T) \tag{34}$$

even though the latter has a smaller bias. This means better predictive power. However, when the bias is too large, the prediction is no longer better than that with greater variance. In other words, by dropping off more variables, a reduced variance is being traded for increased bias. Deletion of a variable makes sense only if a gain of prediction precision (e.g., lower variance) is greater than a loss due to the introduced bias. Since the “true” value is unknown, it is hard to estimate the bias of the solution in order to determine when further reducing of the variance no longer gives prediction improvement. Notice that feature subset selection is the most common regularization technique in pattern recognition and statistics.

Hence, if we include unessential variables in the set of predictors, it will degrade the accuracy of the classification on future data. The use of all features does not guarantee good classification. This implies that proper predictor subset selection is of great importance in the fouling problem and the only reasonable choice must be made based on the best predictive power which, in turn, must be somehow evaluated without having a validation data set (or a correct answer). One method to do that selection, known as complexity penalization of models, is discussed below.

Due to the independent works of Kolmogorov, Solomonoff, and Chaitin [29] on the theory of algorithmic complexity, it is now possible to associate the complexity of objects with the length of their description. The *algorithmic complexity* of an object represented as a binary string is the length of the shortest computer program (i.e., description) that can print that string and halt. The main drawback of this definition is that the so-defined algorithmic complexity is not computable and needs to be approximated to be used for practical applications. This fact does not limit the power of the approach. Note that the number $\pi = 3.14159265\ldots$ is not computable, but its approximations are successfully used everywhere. The only difference is that we can determine the accuracy of our approximation to the number π, but we cannot quantify this about approximations to the algorithmic complexity. One of the possible approximations to the algorithmic complexity of an object is the description length measured as the length of the codeword corresponding to that object. It is well known due to the McMillan-Kraft theorem [30] that if there is a probability distribution defined over a number of objects (or binary strings, x), then there exists a uniquely decodable code with codeword lengths:

$$L(x) = \lceil -\log P(x) \rceil \tag{35}$$

We can refer to $L(x)$ as the description length of the object x (or as the approximate value of its complexity). In other words, we can evaluate the complexity of objects (data or models) by calculating $L(x)$, assuming we know the probability distribution defined over the objects. Another result of using Algorithmic Complexity theory with great importance in regards to the feature selection problem is the definition of *universal distribution* [29], which puts our intention for preferring simpler models into a rigorous form. It is expressed mathematically by:

$$P(x) \propto 2^{-L(x)} \tag{36}$$

It reflects the fact that if a string has appeared due to cause (i.e., a computer program has printed it), the probability that this string is simple (i.e., it has a short description length) is much higher than the probability that it is complex. Therefore, when building a model from this data, one must prefer simpler models because the probability that our data were generated by simpler models (or that our data have shorter descriptions) is much higher. Occam's razor principle states that it is vain to do with more what can be done with less.

The problem of feature selection can be reformulated in terms of probabilistic model classes M_k, each corresponding to the subset models with k predictors:

$$M_k = \left\{ P_\beta \left(Y \mid X_{(1)}, X_{(2)}, \ldots, X_{(k)} \right) \right\}, \tag{37}$$

and denote by $\breve{M}_k$ the model in model class M_k that is optimal in some sense. For many practical applications, we only need to consider the probabilistic model classes. The reason is that deterministic models, given in the form of a functional mapping $f : X \to Y$ with an error function that measures the goodness of fit, are defined as:

$$\delta(y, \hat{y}) = \sum_{t=1}^{n} \delta(y_t, \hat{y}_t) = \sum_{t=1}^{n} (y_t - \hat{y}_t)^2 \tag{38}$$

and can be easily transformed into probabilistic models by using probability distributions induced by the error function [31]. It is easy to show that a maximum likelihood estimate of the parameter for the induced probability distribution minimizes the error function $\delta(y, \hat{y})$. This means that finding the maximum likelihood estimate is equivalent to finding the solution of the original deterministic problem.

Perhaps the simplest way to show how complexity-penalized model selection arises is to use the Bayesian rule where we choose the model that has a maximal posterior probability:

$$P(M_k \mid D) \propto P(D \mid M_k) P(M_k). \tag{39}$$

For many years, the Bayesian inference has been criticized for its using subjective prior probabilities. Now we can use the power of Bayesian inference for selecting a model class among alternative ones using the universal prior distribution on the model classes. This can no longer be claimed to be subjective.

Using the universal distribution (5) over the model classes (6) and taking negative logarithms of both sides of (8) we arrive at a so-called penalized log-likelihood criterion:

$$\begin{aligned} -\log P(M_k \mid D) &\propto -\log P(D \mid M_k) - \log 2^{-L(M_k)} \\ &= -\log P(D \mid M_k) + L(M_k) \end{aligned} \tag{40}$$

The minimum of expression (9) over the model's parameters represents the description length of the data and the model class. According to this criterion, we must choose the model class that results in the shortest description (or has maximum posterior probability). Looking at expression (9), we see that the desired solution represents a tradeoff between the shortest description of the data represented by a model class and the lowest complexity of the model class. In general, any complexity-penalized criterion selects the model to maximize the fit to the data while minimizing the model complexity. The model chosen by using criterion (9) will provide better prediction on future data than those corresponding to longer descriptions. The way of exploiting description lengths for complexity-penalized model selection is simplified for the clarity of explanation. If we follow the reasoning presented in the minimum description

length (MDL) literature [31], minimizing the total length of the code achieved for the data with the help of a proposed model class by using a particular coding scheme, we will arrive at the MDL criterion:

$$MDL(k) \approx -\log P_{\hat{\beta}}(D \mid M_k) + \frac{k}{2}\log n \tag{41}$$

where n is the number of training data points and k is the number of parameters of the model class M_k. This particular form of the MDL criterion is derived using a two-part coding scheme in which the model and the error are coded independently. The MDL principle selects the model class *to minimize the sum of the description length of the model, which increases with model complexity, and the description length of the error, which decreases with model complexity*. To apply the MDL principle, one needs one or more probabilistic model class and data. Then, the proposed model class(es) can be fitted to the data and compared using the description length. The model class corresponding to the shortest description length is chosen as the one that explains the data best. The suggestion of models and model classes for describing data lies beyond the principle. Model classes should be produced by using the creative imagination and engineering judgement of researchers. This principle is simply a selection tool that evaluates models and data and tells which model the best in terms of description length is. For the report, the MDL principle was used to select relevant frequency bands from PSD of the UGW signals described about. Having selected the features, we applied SVM and NNs to study their performance on a reduced set of features in comparison to the full set of features. Results are presented in Table 3.

Analysis of Table 3 reveals that while feature selection could not improve the already perfect SVM classification rate, it made a dramatic difference for NN classification. The recognition accuracy improved to almost 99.8%, practically matching the performance of the LM algorithm, while the standard deviation was reduced to 0.9%. These results demonstrate that feature selection is a powerful technique for NN regularization and should be applied every time when consistency of classification is paramount.

Table 3. Performance of different ML technique with and without feature selection by MDL principle.

Classification Method	Average classification accuracy, %	Standard Deviation of classification accuracy, %
SVM, RBF kernel, full set of features	100	0
SVM, RBF kernel, set of features selected by MDL	100	0
NN with gradient descent training, full set of features	74.4	23.5
NN with gradient descent training, set of features selected by MDL	99.8	0.9

There are other coding schemes that provide valid description lengths of data based on probabilistic model classes [31]. Those schemes have been found to outperform the penalized log-likelihood form of the MDL criterion (41) for some problems. The MDL criteria operates by code lengths and can be used to compare any type of model. Another advantage is that in the MDL framework, there is no need to assume anything about how the existing data are generated [31]. Although the MDL principle admits the use of prior information in the form of prior distributions, the main idea is to use available prior information for suggesting model classes that will be compared with the help of the MDL principle. We favor this approach because description length is correlated with prediction risk. This means that models corresponding to shorter descriptions give better predictions than models corresponding to longer descriptions [26].

For linear regression models, the MDL criterion was derived assuming a two-part code scheme and normally distributed noise ε takes the form [31]:

$$iMDL \approx \frac{n}{2}\log R_c + \frac{1}{2}\log\left|cI + X'X\right| - \frac{1}{2}\log\left|cI\right| \tag{42}$$

where R_c stands for the penalized sum of squared errors $R_c = \delta\left(y, \hat{y}\right) + c\beta\beta'$. The vector of regression coefficients β is calculated to minimize R_c. The regularization parameter c is chosen to minimize the code length (35) [31]. As we can see, the feature selection problem is linked to the selection of regularization parameter, which is discussed in the next section.

3.5 Selection of Regularization Parameters for ML Algorithms

Expression (6) demonstrates that the trade-off between fidelity to the data and fidelity to prior assumptions is controlled by regularization parameter λ, which needs to be selected to optimize the performance of a ML algorithm.

There are two major approaches to regularization parameter selection: deterministic and stochastic. The stochastic approach exploits the statistical nature of the noise component in the response, whereas the deterministic approach completely ignores it. In either approach, there are methods that require different

types of input information for producing a proper value of the regularization parameter for a problem. Figure 17 shows one possible classification of the regularization parameter selection methods (RPSMs). The "Heuristic" and "Error Free" methods do not require an estimate of the noise level in the response; the others do.

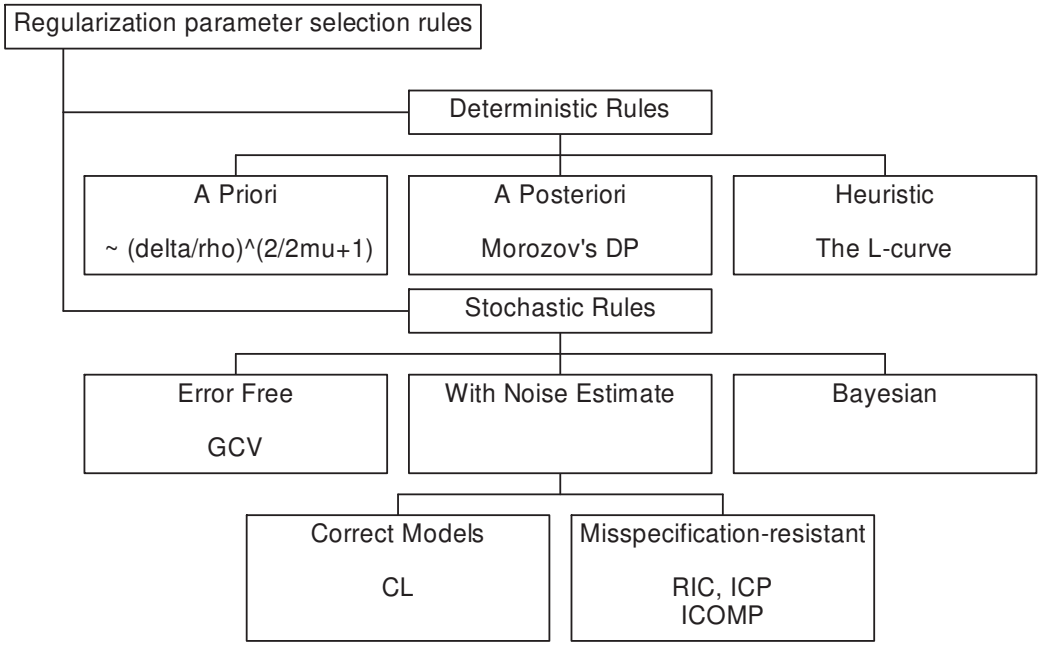

Figure 17. Classification of RPSMs.

3.5.1 Deterministic RPSMs

A priori RPSMs require, as the name implies, *a priori* information about the true solution and/or the true noise level in the response. Since neither is available in practical applications, especially when parameters have no physical interpretation, these methods are of little interest for practical implementations. They are important from the theoretical point of view because they establish optimal convergence rates. A regularization method is convergent when the error between the regularized solution obtained using this method and the true solution goes to zero as the noise in the response goes to zero. The convergence rates are useful in the theoretical analysis of the regularization methods and in comparing different RPSMs. RPSMs with faster convergence would provide more accurate solutions for a given noise level and, thus, are preferable.

Many discrete pattern classification and ML problems can be reduced to the solution of a simple linear equation:

$$Y = Xb + \varepsilon \tag{43}$$

where Y is an $n \times 1$ vector of noisy output signals of a system or process under consideration called the response, X is an $n \times m$ matrix representing n observations or measurements of m independent variables called the predictors, b is a vector of m parameters called the regression coefficients, and ε is an unknown noise vector that represents the measurement error, the modeling error, and the true stochastic noise. In this report, we use formulation (43) to analyze different RPSMs.

3.5.1.1 A priori RPSMs

When the noise level, denoted as δ, is known, and for some $\mu > 0$, $b = \left(X^T X\right)^{\mu} w$, where $\|w\| \le \upsilon$ (i.e., has a source representation), the regularization method is of optimal order with the following *a priori* RPSM (32):

$$\lambda \sim \left(\frac{\delta}{\upsilon}\right)^{\frac{2}{2\mu+1}} \tag{44}$$

This result is for the deterministic setting. The source representation can be seen as a condition on the decay rate of the correlation coefficients ρ_i between Y and u_i. For problem (43) to have a regularized solution, the correlation coefficients ρ_i arranged in decreasing order of the singular values must decay faster than the singular values of $X^T X / n$. For larger μ, this condition becomes more severe. Namely, the correlation coefficients must decay faster than the singular values raised to the $2+4\mu$ power. If this is fulfilled for larger μ, the convergence of the regularized solution to the true one will be faster.

For most real-world applications, neither μ nor υ is known, and, as a result, it is impossible to construct an *a priori* RPSM of optimal order. Therefore, several *a posteriori* RPSMs that depend on the data have been proposed.

3.5.1.2 A posteriori RPSMs

The most widely *a posteriori* RPSM used is Morozov's [33] Discrepancy Principle (MDP). The regularization parameter value is chosen as a solution of the following equation:

$$\left\|Xb_\lambda - Y\right\| \le \delta \tag{45}$$

The regularization parameter λ is chosen such that the corresponding residual (left-hand side of (45)) is less than or equal to the *a priori* specified bound (right-hand side) for the noise level in the response. Since a smaller λ corresponds to less stable solutions, the λ for which the residual equals the specified noise level is chosen. There is no reason to expect a residual less than the noise level. In modeling from

data, a residual less than the noise level in the response corresponds to overfitting, which is a term for learning noise in the training data. The regularization method with λ chosen according to the MDP (45) is convergent and of optimal order [32,33].

To apply MDP, we must have *a priori* knowledge about the noise level in the response. Since the noise level is usually unknown, we use an estimate of the noise level. Unfortunately, MDP is very sensitive to an underestimation of the noise level. This limits its application to cases in which the noise level can be estimated with high fidelity [34]. An improved *a posteriori* method [32] outperforms MDP in that it is of optimal order for a wider range of μ than MDP.

A posteriori RPSMs require the noise level to be either known or reliably estimated. Such a noise level can be hard to obtain. An alternative approach to regularization parameter selection uses noise-level-free RPSMs. Noise-level-free RPSMs are also referred to as heuristic RPSMs. Heuristic RPSMs provide a regularization parameter value without knowledge of the noise level. However, due to the result of Bakushinskii [36], a noise-level-free RPSM cannot provide a convergent regularization method. Therefore, heuristic RPSMs are nonconvergent. Despite that, in practical applications, heuristic RPSMs may demonstrate very good performance in reconstructing the solution of ill-posed problems [34].

The most widely used heuristic method is the L-curve method [34]. In this method, the residual norm is plotted versus the regularized solution norm and the regularization parameter value corresponding to the corner of the L-shape curve is chosen. The corner occurs where the curve has its maximum curvature. The L-curve method has been shown to be nonconvergent [37]. For some problems, it is extremely difficult to locate the corner; for others, the L-curve may have several corners. The L-curve method can also be used in the stochastic setting.

3.5.2 Stochastic RPSMs

In a stochastic setting, a distributional model of the noise component ε in the response is specified. Usually, white Gaussian noise is assumed (i.e., the noise component has a multivariate normal distribution denoted as $\varepsilon \sim N_n\left(0, \sigma^2 I_n\right)$, where ε is a random noise n-vector whose components are independent and normally distributed with zero mean and common variance σ^2). I_n denotes the $n \times n$ identity matrix. A RPSM is obtained so that it minimizes the mean predictive error estimated from the data. Therefore, all RPSMs in the stochastic setting use an estimator of the mean predictive error and select the regularization parameter value, which minimizes the corresponding estimator.

3.5.2.1 Generalized Cross Validation

Probably the most widely used noise-level-free RPSM is Generalized Cross Validation (GCV) [35]. According to this method, the regularization parameter is chosen such that it minimizes the GCV function, as given by:

$$GCV(\lambda) = \frac{\left\| Xb_\lambda - Y \right\|^2 / n}{\left(trace(I - H_\lambda)/n \right)^2}, \tag{46}$$

where $H_\lambda = X\left(X^T X + \lambda I\right)^{-1} X^T$ is called the hat or projection matrix. GCV does not require prior knowledge of the noise level and works with the white Gaussian noise model for the noise component. GCV occasionally fails, presumably due to the presence of correlated noise [35]. GCV can also produce grossly under-regularized solutions [35].

3.5.2.2 Mallows' CL method

Other widely used RPSMs are Mallows' CL method [38] and the Unbiased Risk Estimator (URE) [38], which is similar to CL. CL is derived as an estimator of the mean predictive error, in which the noise level is treated as a nuisance parameter and the components ε_i of the noise vector are assumed to be normally distributed with zero mean and common variance σ^2. CL is given by:

$$CL(\lambda) = \frac{\left\| Xb_\lambda - Y \right\|^2}{n} + \frac{2\sigma^2}{n} trace(H_\lambda) - \sigma^2 \tag{47}$$

CL must be accompanied by either an *a priori* noise level as in the deterministic setting or by a reliable estimate of the noise level. CL is very sensitive to an underestimation of the noise level and may fail to provide a regularization parameter value corresponding to an admissible regularized solution. CL was derived for the white Gaussian noise case and, hence, may not work reliably if that assumption is violated.

3.5.2.3 Information Criteria

GCV and CL methods are defined for uncorrelated Gaussian noise case and cannot be easily extended to more realistic cases. In real applications, the distribution of noise can be non-Gaussian with non-zero values of skewness (i.e., asymmetric) and excess (i.e., narrower or wider than Gaussian). Data can contain outliers and can be generated by a mixture of distributions. The level or variance of the noise may not be stationary, but can vary. The noise may also be correlated. Finally, the statistical model of the noise can

be misspecified; as such, any results obtained without taking this into account can be invalid. None of the above methods can be generalized to any of these conditions.

In order to deal with noise- and model-misspecification and construct misspecification-resistant RPSMs, the information approach that became widely used in statistical model selection due to the works of Akaike [39], Takeuchi [40], Murata [41], and others should be considered.

The main advantage of the information approach is that it accounts for possible functional and distributional misspecifications of the models in a very natural way. While misspecification may not be an issue when solving integral equations, it plays a crucial role in engineering applications based on black-box and data-driven techniques where the very notion of a true model is arguable and usually not discussed, though its existence is assumed. A similar situation arises with econometric models in which misspecification-detection and misspecification-resistant estimation have been extensively used in contrast to engineering. For a detailed treatment of misspecification in modeling and further references on misspecification testing, refer to White [42]. In these situations, methods that are consistent under possible misspecifications are valuable because they automatically guard against the unrealistic assumption of correct model specification. In this report, the information-based criteria, such as the Regularization Information Criterion (RIC) proposed by Shibata [43], was tested as a parameter selection method for NN.

Criteria such as CL and RIC evaluate the generalization (or prediction) error using the training error and an additional term. This additional term penalizes the inaccuracy of parameter estimation and can be interpreted as the effective number of parameters of correctly specified models (for CL) or incorrectly specified models.

With a limited number of observations, penalization of the number of parameters alone becomes inadequate. This additional term cannot be computed exactly because of the dependence on the unknown true distribution and should be estimated from the same data set. As a result, the selected regularization parameter value is often underestimated and produces grossly underregularized or inadmissible solutions. An additional penalization of the parameter estimation inaccuracy, taking into account the interdependencies between the parameter estimates as in the Information Complexity RPSM (ICOMPRPS) proposed in [44], can drastically reduce the risk of regularization parameter value underestimation and make such a choice more suitable for black-box modeling. Such an "overestimation," or more precisely correction, of the inadequate penalization of inaccuracy is beneficial for engineering applications in which the regularization parameter value should be chosen automatically during model building, and there is no means for assessing the proper amount of regularization. For this

report, we tested four regularization parameter selection methods—GCV, CL, ICOMPRPS, and RIC—to select a NN regularization parameter. Results are presented in Table 4.

Table 4. Performance of different regularization parameter selection methods.

Regularization parameter selection method	Average classification accuracy, %	Standard Deviation of classification accuracy, %
GCV	75.2	24.8
CL	75.8	25.3
RIC	76.0	25.1
ICOMPRPS	74.3	23.6

Analysis of Table 4 shows that while optimal regularization parameters improve the average accuracy of an NN classifier, none of the methods show a superior performance. These results demonstrate that the method of regularization is more important for a classifier's performance than an optimally selected regularization parameter.

3.6 Deep Learning NNs

The deep learning paradigm is to use massive recurrent or feedforward NNs to discover latent dependences in large heterogeneous streams of plant data collected with different sensor modalities. Due to large volumes of sensory data collected at NPPs, the first principle models for the majority of the operational regimes at these plants are not feasible. The deep learning system can be calibrated on a plant's data collected during normal operating conditions and subsequently used to detect faults on both system and component levels, such as faulty sensors, small leaks, and component degradation by comparing a system's output with current sensor readings. The deep learning model can also be used for accurate and reliable determination of thermal power through perturbation of different input variables, thus improving a plant's capacity factor. For this report, the deep learning NNs were used to classify signals recorded with high-resolution fiber optic sensors on different piping components. Practical development of deep learning models is a complex multidimensional optimization problem as it requires simultaneous optimization of a large number of parameters, such as a network's architecture and weights, number of input variables, and input training patterns, as well as time lags and temporal variations. Development of deep learning framework for big data analytics will allow more efficient and safe operation of current LWRs.

Deep learning methods have been successfully applied in such diverse fields as speech recognition, computer vision, drug discovery, and bioinformatics. The deep learning architecture shown in Figure 18 uses multiple layers of simple nonlinear information processing units to learn intricate internal structures of stationary, as well as time-dependent, data sets. The fundamental difference of deep learning approach from conventional ML is the lack of feature extraction, feature representation, and feature selection steps.

Feature extraction is the most challenging and time-consuming step in developing conventional ML systems as it requires domain expertise and considerable creativity to extract and represent input information in the form of a multidimensional feature vector to present ML techniques for training or classification [45].

Contrary to this approach, the deep learning paradigm allows processing data in their raw form, thereby bypassing feature extraction, selection, and representation. This is accomplished by using many layers/levels of information-processing elements, each of which perform its own task. Each level processes and presents information at its level of increasing abstraction. For example, the lower level receives, processes, and represents raw data streams from temperature, pressure, flow, vibration, flux, and other sensors present in NPPs. This lowest level of abstraction eliminates outliers and missing values passing this pre-processed data to the next level, which detects linear correlation in the data. The third layer acts upon data obtained from the second layer and detects nonlinear dependencies and so on, thus creating a wholistic model of the underlying process. NPPs are highly complex systems with thousands of parameters that are monitored through different sensor modalities. Due to their complexity, the first principle models holistically describing the operation of an NPP in different regimes are not feasible; however, the data-driven approaches such as deep learning and big data analytics offers an alternative that can be implemented with currently available computational power.

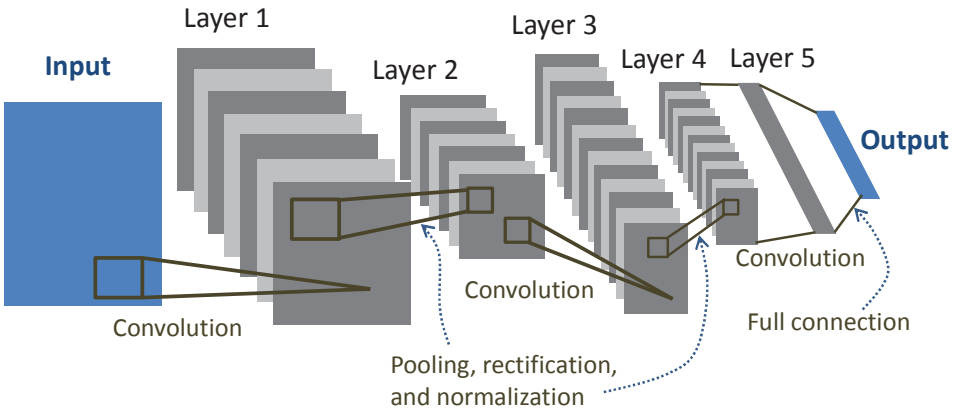

Figure 18. Deep representation learning architecture with five hidden layers.

Deep representation learning is a complex optimization problem as it requires minimization of a cost function with a very large number of adjustable parameters. Deep representation learning usually deploys multilayer perceptron networks in supervised fashion to produce nonlinear mapping of training data into target variables by minimizing quadratic cost function. This minimization is performed by adjusting the parameters of a network, known as weights. In deep representation learning, there are hundreds of millions of weights. This minimization is typically accomplished through variations of a gradient decent

back-propagation algorithm in supervised manner. Due to the nonconvex nature of the optimization problem, there are multiple local minima, thus presenting a challenge of non-unique solutions. Besides the weights optimization, a network's architecture can also be optimized by the number of intermediate layers and number of units in each layer. Also, the number of inputs in the bottom layer can be subject to optimization. Having been trained, the deep learning architectures are validated on some new previously unseen data to ensure that they can generalize well. Deep learning architectures may have up to 20 layers, and with such complexity, they can easily overfit the training data even with hundreds of millions of training examples. To avoid training data overfitting, constrained optimization is used when the cost function is augmented with a penalty term that typically penalizes a model's complexity. This adds one more dimension to the optimization problem to be solved [45].

However, to be able to learn features automatically, a very large number of training data samples need to be presented to the network during the training phase. This is where the availability of big data sets becomes invaluable. All modern enterprises collect data; due to huge increases in computer power and storage volumes, the amount of this data is way beyond a human's capabilities to analyze or simply visualize. NPPs collect data from thousands of sensors with sampling rates in kHz [45]. Recently, attempts were made to tap into this vast information source by the Electric Power Research Institute (EPRI) with their Fleet-Wide Prognostics and Health Management (FW-PHM) system; however, this approach is still based on traditional ML and pattern recognition techniques.

EPRI's system learns patterns of behavior that represent normal operational regimes of systems or equipment. When the currently observed patterns diverge from healthy patterns, the system reports the anomaly as an indicator of potential equipment degradation. However, for the system to operate, it requires a healthy signature database that is used to compare the current pattern against.

A deep NN with a fixed structure represents a parametric family of mathematical functions $\{f(\cdot;\theta|\theta \in R^k)\}$, which are parameterized by the weights of the matrices and bias terms involved in affine transformations. These functions are used as approximations to predict the response variables y_i using the input data x_i. The supervised learning process iteratively modifies the parameter (i.e., weights of the network) to minimize the approximation error $E[\theta]$ on a training data set (N records):

$$E[\theta] = \frac{1}{2}\sum_{i=1}^{N}(f(x_i;\theta) - y_i)^2 \tag{48}$$

If the non-linear functions in the structure of the network are differentiable almost everywhere, which is typically the case, then the gradient of the approximation error $\nabla_\theta E(\theta)$ can be easily derived using the chain rule for differentiation. The efficient procedure for calculating the gradient enables an application of the gradient decent optimization algorithms for network training. These algorithms iteratively update the

weights in the opposite direction of the gradient, until some termination criteria is satisfied. In the standard gradient decent method with the learning rate α, the parameters θ are updated using the gradient calculated on for the whole data set:

$$\theta = \theta - \alpha \cdot \nabla_{\theta} E[\theta] \tag{49}$$

Due to the nonconvex nature of the optimization problem, there are multiple local minima thus presenting a challenge of non-unique solutions. Besides the weights optimization, the network's architecture also can be optimized, such as the number of intermediate layers and units in each layer. Also, the number of inputs in the bottom layer can be subject to optimization. Having been trained, the deep learning architectures are validated on some new previously unseen data to ensure they can generalize well. Deep learning architectures may have up to 20 layers, and with such complexity, they can easily overfit the training data even with hundreds of millions of training examples.

To avoid training data overfitting, constrained optimization is used when the cost function is augmented with a penalty term, which typically penalizes a model's complexity. This adds one more dimension to the optimization problem to be solved [46].

The standard gradient decent method is subject to a variety of numerical issues. Mainly, the iterations defined in equation (49) can be very slow, since this requires going over all of the records in the data set. The Stochastic Gradient Decent (SGD) partially alleviates these issues by approximating the gradient using a small random subset S of the training data, leading to a following update rule [47]:

$$\theta = \theta - \alpha \cdot \frac{1}{2} \sum_{i \in S} \nabla_{\theta} (f(x_i; \theta) - y_i)^2 \tag{50}$$

There are a number of extensions of the SGD method, which improve the convergence speed of the optimization and numerical stability (Nesterov's Accelerated Gradient Decent [48], Adagrad [49], and Adam [50]).

4. PIPING MONITORING USING DISTRIBUTED FIBER ACOUSTIC SENSOR AND ARTIFICIAL INTELLIGENCE BIG DATA ANALYTICS

4.1 High SNR Phase-Sensitive Distributed Acoustic Sensing

Fiber-optic distributed acoustic sensing (DAS) with phase-sensitive optical time-domain reflectometry (ϕ-OTDR) is a powerful distributed sensing technology used to detect acoustic and vibration signatures for a wide array of applications. ϕ-OTDR offers high multiplexing capability using low-cost standard telecommunication fibers as a sensing medium. By using a narrow linewidth laser source and coherent detection, ϕ-OTDR systems can achieve highly sensitive acoustic detections with high-spatial resolutions.

In a ϕ-OTDR system, the performance of the sensing dynamic range, spatial resolution, and sensitivity are governed by the system SNR, *which is severely limited by the low intrinsic Rayleigh scattering coefficient of the optical fiber*. Data processing techniques, such as moving average and wavelet transform, could improve an SNR to an ϕ-OTDR system, but with significant drawbacks leading to low frequency responses. The femtosecond laser fabrication is schematically shown in Figure 19.

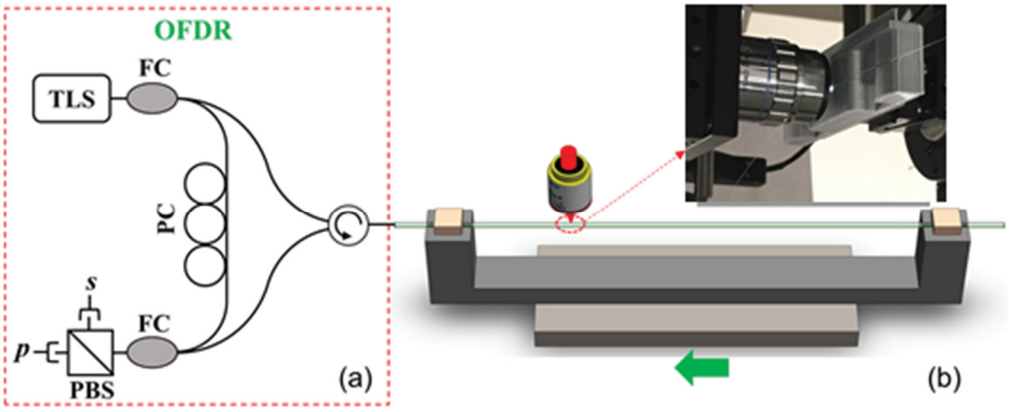

Figure 19. Schematic sketch of the Rayleigh Enhancement setup. (a) Optical Frequency Domain Reflectometry (OFDR) system (LUNA OBR 4600 with internal components—TLS: tunable laser source; FC: fiber coupler; PC: polarization controller; and PBS: polarizing beam splitter). (b) A schematic sketch of the ultrafast laser irradiation on optical fibers.

The ultrafast laser system at the University of Pittsburgh consists of a Coherent MIRA-D Ti: sapphire seed oscillator and a RegA 9000 regenerative amplifier operating at 800 nm with a repetition rate of 250 kHz. The pulse width was adjusted to 300-fs. A cylindrical telescope was used to shape the laser beam and control the shape of the focal volume. Oil immersed objectives (80×) were used to process cylindrical shaped fibers, as shown in the inset of Figure 19. The fiber being irradiated is also interrogated using a commercial OFDR interrogator (LUNA OBR4600). A roll-to-roll fiber handling setup is available, which allows continuous processing of the optical fiber. Through the optimization of laser pulse

width, pulse energy, and laser writing speed, nanograting can be formed inside the fiber core by the focused laser pulse to locally enhanced Rayleigh scattering with appropriate length and Rayleigh enhancements.

Figure 20 shows a Rayleigh back-scattering profile along a 60-meter long sensing fiber (i.e., standard telecom fiber) enabled by fs-laser. Each section is 5-mm long with at least 30-dB back-scattering signal enhancement, which is >1000 times stronger than that of the pristine fiber shown in Figure 20. The Rayleigh enhancement is wavelength independent, which responds to all interrogation wavelength. The strength, location, and section length are all controllable, which can flexibly be realized by our laser processing system. The Rayleigh enhancement will dramatically improve SNR of ϕ-OTDR systems dramatically and make high-precision measurements possible. Phase changes induced by acoustic/vibration exerted on section fibers between two adjacent Rayleigh enhanced sections can be captured and demodulated using a ϕ-OTDR system developed by our research team. The demodulation system is schematically shown in Figure 21.

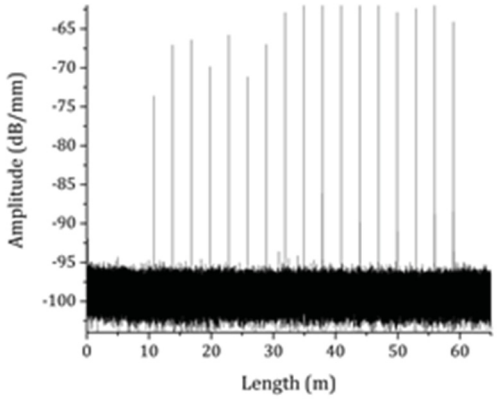

Figure 20. Enhanced back-scattering in standard fibers enabled by fs-laser.

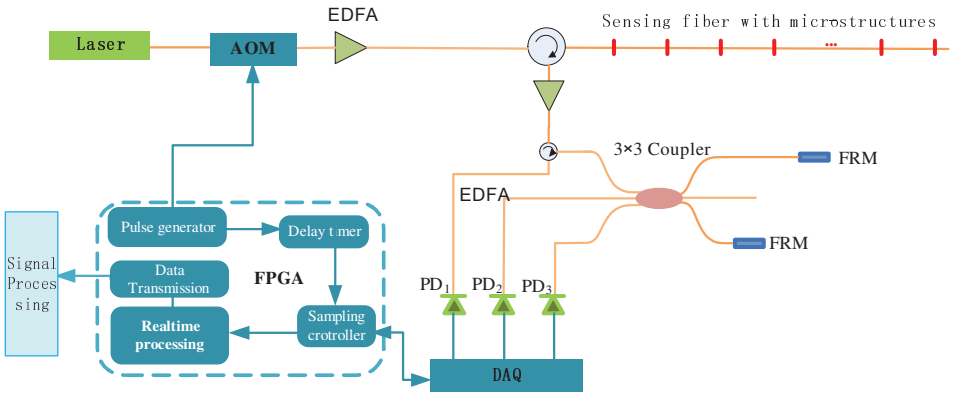

Figure 21. Schematic diagram of the φ-OTDR sensing system enhanced by microstructures.

A single wavelength light from a narrow-line laser is modulated to nanosecond pulses by an acoustic-optical-modulator (AOM). The pulses are then amplified by an optical amplifier, an Erbium-doped fiber amplifier (EDFA), and launched into a sensing fiber. The reflected pulses from the sensing fiber, which are enhanced by 17 locally reflective points, return to a 3×3 coupler through a circulator. The pulse gets through the long and short arms of a 3×3 coupler, respectively, and are both reflected by faraday rotator mirrors (FRMs). If the path-match condition is satisfied, the 3×3 coupler and FRMs will comprise a balanced Michelson interferometer. The interference signals are collected by photodetectors. The 3×3 demodulation method is used to obtain phase changes in the fiber. The three output signals have a 120° phase shift through the 3×3 coupler, which can be described as:

$$I_k = D + I_0 \cos[\varphi(t) - (k-1)(\frac{2\pi}{3})] \tag{51}$$

where k(k=1,2,3) is the output number, D is the average of the output light intensity, and I_0 is the peak intensity of interference signals; $\varphi(t) = \phi(t) + \psi(t)$, where $\phi(t)$ and $\psi(t)$ are respectively phase shifted, caused by the signal to be detected and environmental noise. The output signal after the demodulation algorithm is:

$$V_{out} = \sqrt{3}\varphi(t) = \sqrt{3[\phi(t) + \psi(t)]} \tag{52}$$

The phase changes, caused by vibration, are quantitatively detected for further analysis. The University of Pittsburgh research team has strong expertise in developing photonic instrumentation. The team has developed a preliminary ϕ-OTDR system based on a high-speed A/D data acquisition card and high-speed FPGA data processing system for real-time data processing. The prototype system is shown in Figure 22.

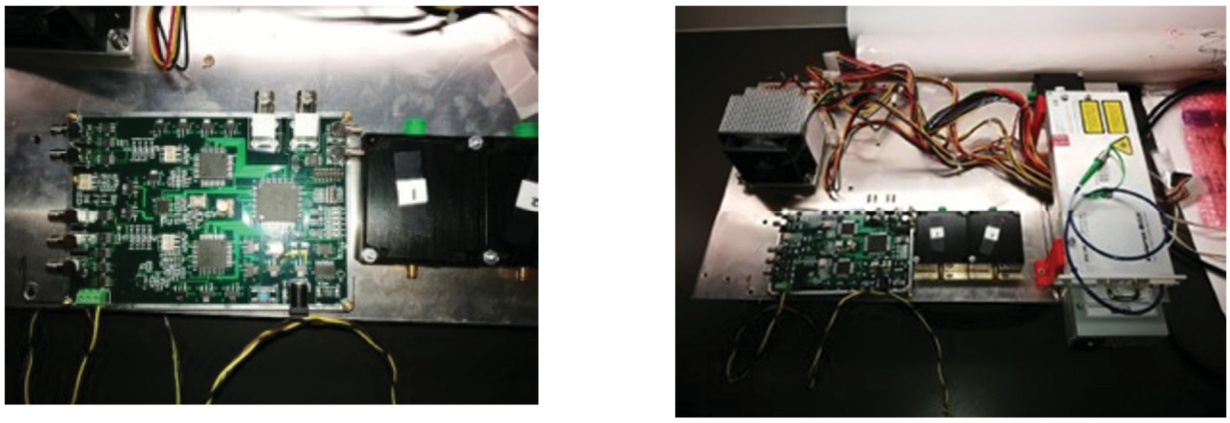

Figure 22. Photograph of custom developed circuit boards for ϕ-OTDR system prototype.

4.2 Preliminary Results Obtained with High SNR Phase-Sensitive Distributed Acoustic Sensing

Using the current femtosecond laser sensor fabrication method (see Figures 19 and 20) and the DAS prototype (see Figures 21 and 22), we performed preliminary distributed acoustic sensing experiments on the pipeline to explore the possibility of identifying defects on certain pipe locations, which were difficult to detect. The iron pipe used for these experiments is shown in Figure 6. The inner diameter (ID) of the pipe was 4-in. with a wall thickness of ½-in. It consists of three sections including two 5-ft. long straight sections connected by a 90° elbow. As was mentioned before, the traditional UGW detection scheme cannot effectively detect and identify defects in complex structures, such as elbows, due to their complex ultrasonic echo-features. However, the distributed fiber sensor technology can effectively enhance the ultrasonic inspection scheme. As shown in Figure 23, a 20-meter long fiber pipe with 14 Rayleigh enhanced points are inscribed using an ultrafast laser. These 14-Rayleigh enhanced points form seven sections (two enhanced points per section). This will enable us to perform ultrasonic measurement at multiple locations along the pipeline using a one-fiber/one-fiber feedthrough configuration, as is shown in Figure 23.

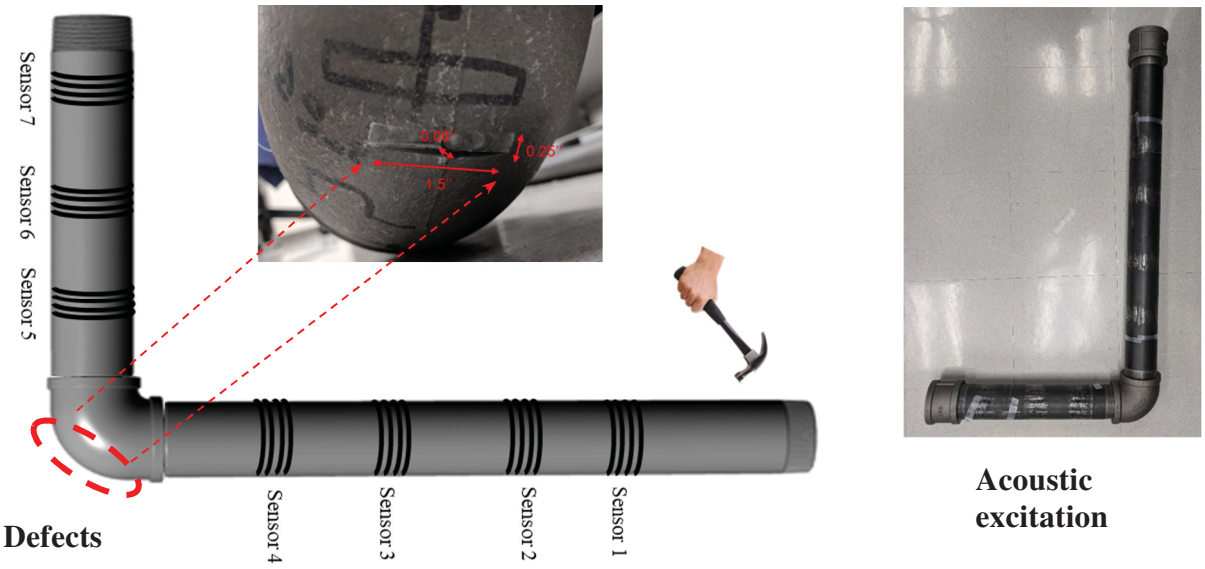

Figure 23. (left) Schematic of pipeline monitoring of defects on elbow using distributed acoustic sensors, and (right) photograph of the experimental setup.

To simulate corrosion defects, a 0.08" (2-mm) deep trench (at the deepest) was cut into the steel elbow, as is shown in Figure 23. Acoustic excitation was generated using a specialized acoustic hammer. The acoustic excitation and its frequencies can be changed using different hammer heads. In this experiment, four types of hammer heads were used including rubber, plastic, aluminium, and steel. To

simulate the different pipeline installation scenarios, acoustic signatures were collected on three different pipeline situations including:

- Situation 1 – pipeline with good elbow
- Situation 2 – pipeline with defected elbow
- Situation 3 – pipeline with good elbow, but elbow was loosely connected to a straight pipe.

The representative acoustic signatures detected by these three situations are shown in Figure 24. It is evident that all seven distributed acoustic sensors can detect an acoustic signal generated by the hammer. Through detailed analysis of the acoustic signal and its arrival times, it will provide more information on pipeline structural health along the entire pipeline with better spatial resolution than PZT-based acoustic sensors monitored in one location. This is particularly advantageous for structural health monitoring of complex pipe structures (e.g., elbow in this case). The multiple fiber sensors, as shown in Figure 23, can be interrogated using the one-fiber/one-fiber feedthrough method, greatly reducing wiring complexities and installation costs in the process.

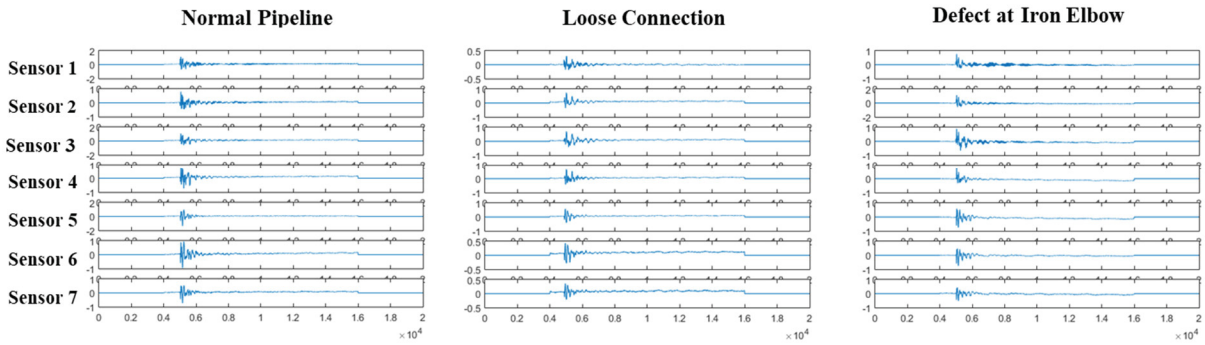

Figure 24. Acoustic signal measured by seven fiber sensors for three different situations. The signal was generated by an acoustic hammer using a rubber head.

However, similar to PZT-based sensors, the acoustic signal detected by the fiber sensors, as shown in Figure 24, is highly susceptible to where fiber sensors are mounted on pipelines, their relative locations, and the particular acoustic modes generated by the hammers. This indeed will pose great challenges for data analytics using a deterministic signal processing approach. Further, distributed fiber sensors will generate a much larger set of data than that produced by PZT-based sensors installed in one location. In addition, corrosion-induced defects come in different forms, sizes, and severities. The acoustic signals they generate are subtle, which are showcased in Figure 7, while the acoustic signals generated by the three different scenarios above are almost undistinguishable.

Given the challenges with data analysis, our research group performed our data analysis using an artificial intelligence big data approach. In this preliminary work, we use deep neural networks (DNNs) to

perform data analysis and feature identifications. Our focus in this preliminary work was the Convolutional Neural Network (CNN), which is a type of NN capable of extracting multiple local features from layer to layer. The usage of convolution in the CNN is especially effective to handle data collected from distributed sensors and reveal spatial dependencies of the data. The CNN has achieved significant success in image processing and other domains of artificial intelligence.

Basic processing of the raw data makes it easier for the CNN to handle the data. A Fast Fourier transform (FFT) was used to obtain the frequency components of the data, giving the CNN direct access to the global properties of the signals. Meanwhile, by applying low-pass filtering to eliminate irrelevant high-frequency components and sync-filtering, which corresponds to zero-padding in the time-domain, it was possible to smooth the representation in the frequency domain. Figure 25 shows the architecture of the CNN. The input is the first 512 frequency components of all seven sensors extracted from FFT. The nonlinear activation function is via a rectified linear unit (ReLU). Max pooling is used between each layer to reduce the data in each channel. After each layer, the number of channels (i.e., the number of features extracted) increases. The output stage is a softmax classification layer fully connected to the previous layer.

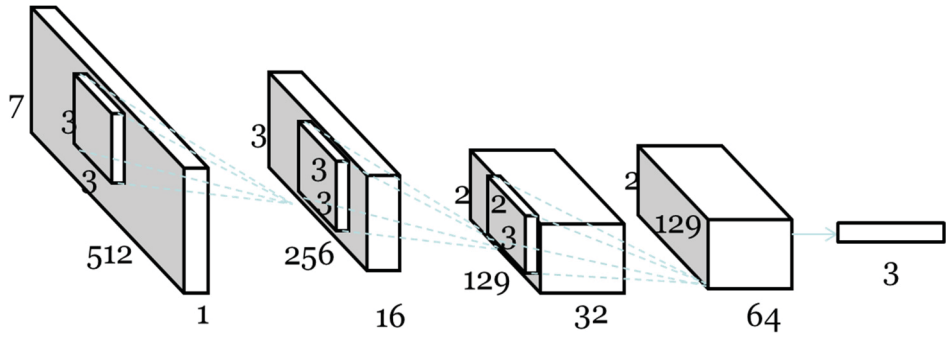

Figure 25. Architecture of CNN used for defect recognition.

To use this ML algorithm to identify defects, we used an acoustic hammer to generate 50 sets of data on each situation as outlined before (e.g., normal, defected, loose elbow connections) by each hammer head. In total, 600 data sets were produced. These data were generated on different days at different times by different people. This was intentionally done to examine the efficacy of the CNN method. In this preliminary study, we used supervised learning for feature identification and extraction. Ten sets of data for each situation were used for training of the CNN. Table 5 shows the CNN pattern recognition results after the 10-set data was used for supervised learning.

Table 5. CNN classification results.

Hammer head	Classification Accuracy
Aluminum	94.29%
Plastic	92.75%
Rubber	83.62%
Steel	76.60%

Table 5 shows the identification accuracy using CNN for acoustic data generated by the four different hammer heads. The signal generated by the aluminium head achieves the highest accuracy of 94.29% to successfully identify the three situations (e.g., normal elbow, defected elbow, loose connection). The signal generated by the plastic and rubber hammer heads achieved slightly lower successful rates, probably due to the weaker signal. The lowest identification successful rate was found using a steel hammer head, probably due to a resonating issue produced by a steel hammer head. The results presented in Table 5 highlight success and good potential for integrating a distributed sensor and artificial intelligence data analytics approach.

5. CONCLUSIONS AND RECOMMENDATIONS

The ICA algorithms seem to be a natural fit in separating corrosion defect signals from coherent noise in the context of temperature changes and constant frequency content of the defect signal. Due to interference, multiple GW sensors receive combined information from various sources. This creates observed signals that align very well with the fundamental ICA model, making ICA a logical choice for filtering GW data. However, currently available ICA algorithms have serous limitations when applied to real-world industrial data in that they only deal with linear mixtures and they require that the number of sources do not exceed the number of sensors. While reducing the time window available for processing will reduce the number of sources in that window, in practice, the number of sources is unknown. Practically, useful ICA algorithms should be able to deal with an unknown number of sources and nonlinear and convolutional mixtures. In future work, we plan to process data from axial sensors from the same data set and study the influence of the number of sensors on the quality of separation. Processing axial data from the same data set will provide an opportunity to work with more sources, since the number of axial sensors is higher.

Because ICA is invariant to the ordering of the ICs, automatic algorithms will be needed to label ICs that can be attributed to noise, reflections from engineering features, and reflections from corrosion. Since ICA does not perform dimensionality reduction, all ICs initially have to be considered useful and their ranking needs to be performed based on some other criteria than variance.

One of the benefits of having GW systems permanently installed is the ability to track defect growth by subtracting current reflections from some baseline reflections taken earlier. The results in this report show that ICA has the potential to improve SNR, thus making it a viable tool for tracking defect growth. In future work, we plan to apply ICA to baseline subtraction to investigate opportunities for defect development monitoring. The GW system measurements were in general consistent with the measurement provided by an UT system in terms of locating corrosion activity on the shell. However, the GW system was unable to differentiate defect growth during the monitoring period.

Also, it would be beneficial to study the ability of ICA to deal with shadowing since it is one of the types of interference present in GW signals.

The ICA technique is also only one of several possible techniques for processing GW monitoring data. A through comparison of different signal processing and pattern recognition techniques would help to elucidate when each method is most suitable. Such a comparison will also be a subject of future work.

Many engineering problems are ill-posed. The failure to realize this fact can lead to unsuccessful attempts to build a data-driven method that is reliable and stable. An ill-posed problem is not solvable by

conventional methods because the assumptions under which the methods were derived are violated. For example, it is impossible to build a pattern classification system using the ordinary least squares (OLS) method. The OLS solution in the case of highly collinear predictors is extremely unstable and hypersensitive to small perturbations and particular realizations of the noise component. This is exactly the opposite property a data-driven method should possess to be of a practical value.

Results presented in this report demonstrate that to be practically useful in an online monitoring system, NN needs to be properly regularized either by training or variable selections. Our findings indicate that the most promising methods of regularization for backpropagation NN are the LM algorithm method, Bayesian regularization, and cross-validation. These three methods produced the highest classification accuracy with the lowest variance. In addition, variable selection proved to be a viable alternative to these methods as it matches accuracy and consistency of the above-mentioned techniques. Variable selection, however, requires an additional step when developing a pattern recognition system, which may present a problem if computational resources are not available. The selection of the regularization parameter proved to have a minor influence on the performance of the pattern recognition system; as such, any of the tested methods can be used in practice. On the other hand, SVM demonstrated an outstanding performance in terms of both classification accuracy and stability. SVM was the only method to achieve 100% classification accuracy regardless of the training patterns used. It should be noted that SVM does not use random initialization for the optimization problem, which partly explains its perfect stability. However, prior to training, SVM requires the selection of kernels, which may lead to some variability in classification results. The SVM's stability with respect to kernel selection will be a subject of future studies. Overall, SVM should be a method of choice for online pattern recognition systems due to its classification performance and consistency.

For fiber optic high-resolution sensors, the deep learning CNN produced results that are comparable with shallow NN classification accuracy; however, they do not match the accuracy of SVM. Based on the results obtained in these studies, advances signal processing and data analytics have the potential to significantly improve the detection capabilities of piping monitoring systems in NPPs.

6. REFERENCES

1. World Nuclear Association, 2017, *Nuclear Power Economics and Project Structuring*, January 2017. [Online]. Available at: http://www.world-nuclear.org/getmedia/84082691-786c-414f-8178-a26be866d8da/REPORT_Economics_Report_2017.pdf.aspx. Accessed September 19, 2018.
2. Nuclear Energy Institute, 2016, *Delivering the Nuclear Promise: Advanced Safety, Reliability and Economic Performance*, February 2016. [Online]. Available at: http://www.bhienergy.com/assets/Delivering-the-Nuclear-Promise.pdf. Accessed September 19, 2018.
3. Imperial College London, 2018, "Area monitoring using ultrasonic guided wave." [Online]. Available at: http://www.imperial.ac.uk/non-destructive-evaluation/research/inspection-and-monitoring/area-monitoring-using-ultrasonic-guided-wave/. Accessed September 19, 2018.
4. Alleyne, D., Vogt, T., and Pavlakovic, B., 2016, "Monitoring corrosion with guided waves," Electric Power Research Institute Workshop for Structural Health Monitoring of Passive Components, EPRI Charlotte Office, April 13–14, 2016.
5. Lowe, M. J. S., Alleyne, D. N., and Cawley, P., 1998, "Defect detection in pipes using guided waves," *Ultrasonics*, 36(1–5), 147–154.
6. Li, J., 2005, "On circumferential disposition of pipe defects by long-range ultrasonic guided waves," *J. Pressure Vessel Technol.*, 127(4), 530–537.
7. Coey, J. M. D., 2010, *Magnetism and Magnetic Materials*, Cambridge University Press, Trinity College, Dublin, Ireland.
8. Light, G., and Puchot, A., 2011, *Magnetostrictive Sensor (MsS) Structural Health Monitoring System for Feedwater Heat Exchanger Shell*, Tech. Report No. P27696/C13153, Electric Power Research Institute, Palo Alto, California, USA.
9. Puchot, A., 2013, *Continued Monitoring and Analysis of MsS Data Collected on a Heat Exchanger Shell 13A*, Tech. Report No. 18.18166.02, Southwest Research Institute, San Antonio, Texas, USA.
10. Joliffe, I. T., 1986, *Principal Component Analysis*. Springer-Verlag, New York, New York, USA.
11. Hyvarinen, A., 1999, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks*, 10(3), 626–634.
12. Hyvarinen, A., 1999, "Survey on independent component analysis," *Neural Computing Surveys,* 2, 94–128.
13. Hyvarinen, A., Karhunen, J., and Oja, E., 2001, *Independent Component Analysis*. John Wiley & Sons, New York, New York, USA.
14. Huber, P. J., 1985, "Projection pursuit," *Annals of Statistics*, 13(2), 435–475.
15. Loue, M., and Cauley, P., 2006, *Long range guided wave inspection usage - Current commercial capabilities and research directions*, 29 March 2006, Department of Mechanical Engineering, Imperial College London. [Online]. Available at: http://www3.imperial.ac.uk/pls/portallive/docs/1/55745699.PDF. Accessed September 19, 2018.
16. Cauley, P., Cegla, F., and Galvagni, A., 2012, "Guided waves for NDT and permanently-installed monitoring," *Insight*, 54, 594–601.
17. Eriksson, J., Gulliksson, M., Lindsröm, P., and Wedin, P.-Å., 1996, *Regularization tools for training feed-forward neural networks part I: Theory and basic algorithms*, Tech. Report No. UMINF-96.05, Department of Computing Science, Umeå University, Umeå, Sweden.

18. Bishop, C. M., 1996, *Neural Networks for Pattern Recognition*. Oxford University Press, New York, New York, USA.

19. MacKay, D. J. C., 1992, "Bayesian interpolation," *Neural Computation*, 4(3), 415–447.

20. MacKay, D. J. C., 1992, "A practical Bayesian framework for backpropagation networks," *Neural Computation*, 4(3), 448–472.

21. Hertz, J., Krogh, A., Palmer, R. G., 1991, "Introduction to the theory of neural computation," Lecture Notes, Volume I, in the *Santa Fe Institute Studies in the Sciences of Complexity*. Addison-Wesley, Redwood City, California, USA.

22. MacKay, D. J. C., 1995, "Bayesian methods for neural networks: Theory and applications," Course notes for Neural Network Summer School. University of Cambridge, Programme for Industry. [Online]. Available at: http://www.inference.org.uk/mackay/cpi4.pdf. Accessed September 19, 2018.

23. Cherkassky, V., and Muller, F., 1998, *Learning from Data: Concepts, Theory, and Methods*, John Wiley & Sons, Inc., New York, New York, USA.

24. Gull, S. F., 1989, "Bayesian inductive inference and maximum entropy," in *Maximum Entropy and Bayesian Methods in Science and Engineering, Volume I: Foundations*, Erickson, G. J., and Smith, C. R. (eds.), pp. 53–74. Kluwer Academic, Dordrecht, The Netherlands.

25. Vapnik, V., 1998, *Statistical Learning Theory*, Wiley, New York, New York, USA.

26. Vapnik, V., 1995, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, New York, USA.

27. Linhart, H., Zucchini, W., 1986. Model Selection, Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, Inc.

28. Miller, A.J., 1990. Subset selection in regression, Monographs on Statistics and Applied Probability, Chapman and Hall.

29. Li, M., Vitanyi, P., 1997. An introduction to Kolmogorov complexity and its applications, Springer-Verlag New York, Inc.

30. Gallager, R.G., 1968. Information theory and reliable communication, Wiley.

31. Rissanen, J. J., 1996, "Fisher information and stochastic complexity," *IEEE Trans. Inf. Theory*, 42(1), 40–47.

32. Engl, H. W., Hanke, M., and Neubauer, A., 2000, *Regularization of Inverse Problems*. Kluwer Academic, Dordrecht, The Netherlands.

33. Morozov, V. A., 1984, *Methods for Solving Incorrectly Posed Problems*, Springer-Verlag, New York, New York, USA.

34. Hansen, P. C., 1998, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, USA.

35. Wahba, G., 1990. *Spline Models for Observational Data.* Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, USA.

36. Bakushinskii, A. B., 1984, "Remarks on choosing a regularization parameter using the quasi-optimality and ratio criterion," *USSR Comp. Math. Math. Phys.*, 24(4), 181–182.

37. Vogel, C. R., 1996, "Non-convergence of the L-curve regularization parameter selection method," *Inverse Problems*, 12(4), 535–547.

38. Mallows, C. L., 1973, "Some comments on C_p," *Technometrics*, 15(4), 661–675.

39. Akaike, H., 1973, “Information theory and an extension of the maximum likelihood principle,” in *Proceedings of the 2nd International Symposium on Information Theory*, Petrov, B. N., and Csaki, F. (eds.), pp. 267–281. Akademiai Kiado, Budapest, Hungary.

40. Takeuchi, K., 1976, “Distribution of information statistics and criteria for adequacy of models,” *Mathematical Sciences*, 153, 12–18 (in Japanese).

41. Murata, N., Yoshizawa, S., and Amari, S., 1994, “Network information criterion–determining the number of hidden units for an artificial neural network model,” *IEEE Trans. Neural Netw.*, 5(6), 865–872.

42. White, H., 1994, *Estimation, Inference, and Specification Analysis*, Cambridge University Press, John Wiley & Sons, Inc., New York, New York, USA.

43. Shibata, R., 1989, “Statistical aspects of model selection,” in *From Data to Model*, Willems, J. C. (ed.), pp. 215–240. Springer-Verlag, New York, New York, USA.

44. Urmanov, A. M., Gribok, A. V., Bozdogan, H., Hines, J. W., and Uhrig, R. E., 2002, “LETTER TO THE EDITOR: Information complexity-based regularization parameter selection for solution of ill-conditioned inverse problems,” *Inverse Problems*, 18(2), L1–L9.

45. Krizhevsky, A., Sutskever, I., and Hinton, G. E., 2012, “ImageNet classification with deep convolutional neural networks,” in *NIPS’12, Proceedings of the 25th International Conference on Neural Information Processing Systems – Volume 1*, pp. 1097–1105. Curran Associates, Inc., Red Hook, New York, USA.

46. Gribok, A. V., Hines, J. W., Urmanov, A. M., and Uhrig, R. E., 2002, “Regularization of ill-posed surveillance and diagnostic measurements,” in *Power Plant Surveillance and Diagnostics*, Ruan, D., and Fantoni, P. F. (eds.), pp. 299–317, Physica-Verlag, Heidelberg, Germany.

47. Robbins, H., and Monro, S., 1951, “A stochastic approximation method,” *Annals of Mathematical Statistics*, 22(3), 400–407.

48. Nesterov, Y. E., 1983, “A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$,” *Doklady AN SSSR* (translated as *Soviet Mathematics Doklady*), 269(3), 543–547.

49. Duchi, J., Hazan, E., and Singer, Y, 2011, “Adaptive subgradient methods for online learning and stochastic optimization,” *J. Mach. Learn. Res.*, 12, 2121–2159.

50. Kingman, D. P., and Ba, J. L., 2014, “Adam: A method for stochastic optimization,” *Computing Research Repository (CoRR)*. [Online]. Available at: http://arxiv.org/abs/1412.6980. Accessed September 19, 2018.