Explainable Models for Multivariate Time-series Defect Classification of Arc Stud Welding

Sadra Naddaf-Sh[1], M-Mahdi Naddaf-Sh[1], Maxim Dalton[2], Soodabeh Ramezani[2], Amir R. Kashani[2], Hassan Zargarzadeh[1]

[1] *Phillip M. Drayer Electrical Engineering Department, Lamar University, Beaumont, Texas, USA*
snaddafsharg@lamar.edu
mnaddafsharg@lamar.edu
h.zargar@lamar.edu

[2] *Artificial Intelligence Lab, Stanley Black & Decker, USA*
maximdalton@sbdinc.com
soodabeh.ramezani@sbdinc.com
amir.kashani@sbdinc.com

ABSTRACT

Arc Stud Welding (ASW) is widely used in many industries such as automotive and shipbuilding and is employed in building and jointing large-scale structures. While defective or imperfect welds rarely occur in production, even a single low-quality stud weld is the reason for scrapping the entire structure, financial loss, and wasting time. Preventive machine learning-based solutions can be leveraged to minimize the loss. However, these approaches only provide predictions rather than demonstrating insights for characterizing defects and root cause analysis. In this work, an investigation of defect detection and classification to diagnose the possible leading causes of low-quality defects is proposed. Moreover, an explainable model to describe network predictions is explored. Initially, a dataset of multi-variate time series of ASW utilizing measurement sensors in an experimental environment is generated. Next, a set of techniques to leverage synthetic measurements, reference, and residual signals, and generate a residue dataset, are proposed. Finally, the architecture of classification models is optimized and by Bayesian black-box optimization methods to maximize their performance. Our best approach reaches an F1 score of 0.84 on the test set. Furthermore, an explainable model is employed to provide interpretations on per class feature attention of the model to extract sensor measurement contribution in detecting defects as well as its time attention.

Keywords: Time Series, Arc Stud Welding, Machine Learning, Deep Learning, Defect Classification, ICA, Explainable AI, Bayesian Optimization, Root Cause, Data-Centric AI.

https://doi.org/10.36001/IJPHM.2023.v14i3.3125

1. INTRODUCTION

Arc Stud Welding (ASW) is a standard process in many industrial production lines, such as the automobile industry, and shipbuilding(Samardžić, Klarić, & Siewert, 2007), where a metal fastener called Stud is welded to a workpiece with an electric arc (Al-Sahib, Ameer, & Ibrahim, 2009). The process includes heat transfer, mass transfer, metallurgical reaction, element diffusion, micro-structure change, and variation of mechanical properties, all of which make arc stud welding a complicated process (Hildebrand & Soltanzadeh, 2014). Additionally, the process is characterized by high currents and short weld times. As the process is used in high-volume manufacturing production lines and is applied for structures with numerous stud welded joints, defective or low-quality welds are inevitable. Therefore, a single defective weld leads to the rejection of an entire large-scale structure which imposes significant extra cost and time on the manufacturing process in high volumes (Al-Sahib et al., 2009; Hildebrand & Soltanzadeh, 2014). Hence, root cause analysis of low-quality or defective welds can result in minimizing disposed structures, financial loss, and optimization of the production line. Similar approaches (Heidarydashtarjandi, Prasad-Rao, & Groth, 2022) can be further applied to optimize costs.

Many factors result in a low-quality or defective weld stud, including weld area surface condition or material, inappropriate weld settings, malfunctioning equipment, and inexperienced welding operators (Chambers, 2001). While many are being avoided by appropriate supervision of the process, predicting a possible defective or substandard stud leads to saving time and costs. Furthermore, detecting precise causes helps experts prevent further issues. While in post-weld Non-destructive testing (NDT) X-ray imaging and automated anal-

ysis are applicable (S. Naddaf-Sh et al., 2022; M.-M. Naddaf-Sh et al., 2021), such techniques are not practical for complicated structures in automotive where uncounted weld studs appear in hard-to-access areas. Hence, the monitoring of the process and preventive techniques are more applicable. However, such data and the design of the experiment are not publicly available, and most welding setups are only equipped with voltage and current sensors, where no additional sensors are leveraged for monitoring.

In this study, a set of experiments are designed to synthetically generate defective arc stud welds while various sensors monitor and record welding parameters to generate a dataset of multivariate time-series (MTS) measurements for ASW. To enhance the detection process, a set of reference signals are logged, and synthetic measurements are generated with residual signals to shape a residue dataset. Finally, enhancements of the framework are empirically shown, and a comparative analysis of the methods for the classification of defects is investigated and presented.

Contributions of this work are as follows:

- An extended dataset of experiments is generated based on four sensor measurements during ASW for the purpose of monitoring the operation.
- A new algorithm to calculate the mean approximation of time-series signals and signal residuals is proposed, leading to the enhancement of the overall processing and classification model performances.
- A set of synthetic time-series measurements (e.g., Power) are generated and leveraged to provide enhanced measurements for models and to improve defect (root cause) classification results.
- A set of state-of-the-art classifiers are investigated and optimized for both their architectures and hyperparameters employing Bayesian methods in order to maximize the performance for the mentioned task. In addition, a comparative analysis of their performance is reported.
- Performance of an explainable deep MTS classification model is investigated and optimized to provide interpretation on how the model decides on features to classify signals as well as its time attention.

The remainder of this paper is organized as follows. Section 2 described related works in this scope. Section 3 gives an overview of the terms in the paper, the applied method for synthetic measurement and residual generation steps as well as network architectures. In section 4, the data preparation and dataset, training, evaluation, as well as experimental results, are explained in detail. Finally, in section 5 a discussion on performance is proposed, and in section 6, the paper is concluded, and future improvements are presented.

2. RELATED WORK

In the past decades, few works are published for the enhancement of the ASW process with AI-powered technologies. In (Samardžić et al., 2007), an offline evaluation of parameter distribution during ASW is performed for seven trials with the purpose of common defect detection. Several experiments with various welding conditions, such as a surface with primer or rust, were designed, and parameters such as voltage, current, power, and resistance were measured. Results based on statistical analysis illustrate that, in most abnormal welding conditions, increased variation for voltage and current for fewer cases, e.g., surface with primer, is observed. The study is conducted by manual analysis of measurements.

In (Al-Sahib et al., 2009), quality monitoring and stability of the ASW process are investigated. Moreover, methods for real-time NDT for defect prediction are presented. Researchers present two neural net designs for monitoring weld quality. Welding Time, Welding Current Range, Workpiece Thickness, and Stud Diameter are used as inputs for both networks. One network has two outputs of Welding Current Peak Value and Torque at Failure, and the second network has a single output of visual Inspection (either defective or not). For both networks, a similar training set with only twenty training data was used. In the end, the relationship between the current peak and welding condition was observed and proved the advantage of utilizing the current peak for monitoring purposes. Moreover, employing these networks show the reading of additional weld quality-related parameters. Finally, all the approaches are tested on a tiny dataset with less than twenty samples, and none of them led to imperfection classification. In another work, ASW joint connection status is evaluated by ultrasonic technique (Dong et al., 2019), where A-scan signals are manually analyzed through Wavelet Packet. The authors concluded A-scan characteristics that relate to weak connection zones in stud welds.

More recently, researchers focused on addressing the imbalance issue due to the significant rare occurrence of anomaly cases for the task of fault prediction. In (Zhang, Jha, Laftchiev, & Nikovski, 2019), the authors focus on proposing a new loss function for a recurrent neural architecture that handles imbalance and is specialized for multi-label fault prediction. In another work (Ducoffe, Haloui, & Gupta, 2019), Generative Adversarial Networks (GANs) are employed to learn patterns of anomaly cases in the frequency domain. Later, it is shown that Wasserstein GANs show significantly lower reconstruction error in comparison with variational Auto Encoder.

For supervised time series classification (TSC) and detection of MTS, many studies are done. In (Wang, Yan, & Oates, 2017), Wang et al. presented a baseline for TS classification. Three baselines are presented and compared with older existing ones through 44 benchmark datasets.

The baselines are Fully Convolutional Network (FCN), Residual network (ResNet), and deep multilayer perceptron (MLP). In most cases, these architectures outperformed older methods such as Dynamic time warping (DTW) (Keogh & Ratanamahatana, 2005), The bag-of-features framework (TSBF) (Baydogan, Runger, & Tuv, 2013), and Bag-of-SFA-Symbols (BOSS) (Schäfer, 2015). In (Serrà, Pascual, & Karatzoglou, 2018), a universal encoder is proposed that reaches state-of-the-art performance on MTS benchmark datasets. Finally, In (Fawaz, Forestier, Weber, Idoumghar, & Muller, 2019) a comprehensive review on the classification of both univariate and MTS signals is done by applying both mentioned architectures and a few other designs existing in the literature with competing results on benchmark datasets. The work illustrates that ResNet, FCN (Wang et al., 2017), and Encoder (Serrà et al., 2018) outperform other designs for MTS datasets. These models are appropriate choices for the baseline of our work.

Although many of the available classification methods in the literature have reached significant accuracy on benchmark datasets, a lack of accurate model explainability is discernible. In (Wang et al., 2017; Fawaz et al., 2019) class activation maps (CAM) for univariate datasets are provided. However, in view of architectural limitations, unique designs are required to provide extended explanations for model behavior. In (Assaf, Giurgiu, Bagehorn, & Schumann, 2019), an end-to-end explainable CNN design (MTEX-CNN) is provided to employ gradient-based methods for extracting the time and feature attention of the network on MTS datasets. Fauvel et al. redesigned the network for more accurate performance in prediction results and also significantly has fewer parameters (around ten times less) while boosting the accuracy of explanations of the model called XCM (Fauvel, Lin, Masson, Fromont, & Termier, 2020). In this work, five models of MLP, FCN, ResNet, Encoder, and XCM are optimized and evaluated for ASW defect classification.

3. FRAMEWORK

In an industrial production line, because of the complexity and temporal compression of the process, gathering labeled data on defective welds is impractical. In fact, a non-destructive examination of the entire welded structure is performed afterward. Hence, the design of the experiment is required to carefully simulate similar conditions to reproduce common defective stud welds and generate a dataset of measurements.

In this work, convolutional classification architectures with novel residual signal and reference and synthetic measurement generation of multivariate time-series sensor measurements are utilized, investigated, and optimized. First, signal residuals are calculated based on either an existing reference signal or mean reconstruction of signals using Independent Component Analysis (ICA) whenever a reference signal does not exist. Second, residuals are aligned with original measurements. Moreover, the power and resistance of the process are added by multiplying and dividing the voltages by the corresponding currents, (Samardžić et al., 2007) followed by dynamic time-warping alignment of the resulting signals. Third, Bayesian optimized neural/convolutional architectures are used to perform classification. Finally, network output determines the root causes found in the weld MTS data. The entire system is presented in Figure 1.

In the following, definitions used in the paper are described, then mentioned steps are elaborated in detail.

3.1. Definitions

The technical terms used in this paper are as follows:

- **Time-series**: Consider the time-series of T as a temporal ordered set of n variables sampled with a specific frequency of f and indexed with i to the last value, in the following equation (Fawaz et al., 2019):

  $$T = t[1], t[2], ..., t[n] \tag{1}$$

  which is called univariate time series. On the other hand, an M-dimensional univariate Time-series collection is called Multi-variate Time-series (MTS) and is shown as:

  $$T = T[1], ..., T[M] \tag{2}$$

  Where each T is a univariate time series and M is equal to the number of dimensions (Fawaz et al., 2019).
- **residuals**: A residual of time-series T is calculated by $T' = T - \hat{T}$ where $\hat{T}$ is the mean approximation of T or a reference time-series which accounts for the mean approximation of it.

3.2. Normalization

Each measurement during the process is a T sampled with a frequency of f, and all values per each measurement type and stud type get normalized to zero mean and unit of energy before passing to the next stage using (Goldin & Kanellakis, 1995):

$$\bar{T}[i] = \frac{T[i] - \mu}{\sigma}, \tag{3}$$

where μ is the mean of all measurements with the same type of measurement (e.g., all voltage measurements) and similar stud type, and σ is the standard deviation of the values.

3.3. Residuals

For each univariate measurement, T gets subtracted by either its own mean approximation $\hat{T}$ or an exiting reference measurement for the very measurement type $T' = T - \hat{T}$. For measurements with an existing reference signal, the residual

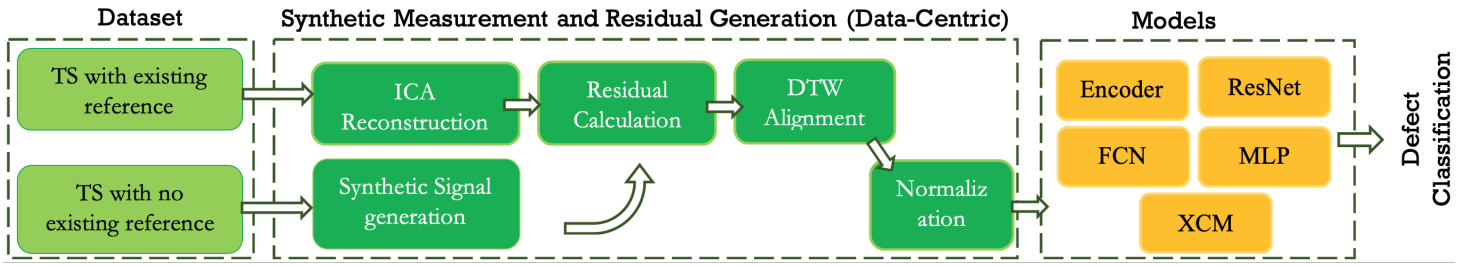

Figure 1. Block Diagram of Defect (root cause) Classification for Arc Stud Welding Process.

calculation is a simple one-by-one subtraction of features for each instance. However, for other signals, a $\hat{T}$ is required. Both ICA reconstruction and signal average were experimentally tested as a mean approximation. However, the mean approximation outperformed the average method in experiments. Next, the mathematical background of ICA is discussed.

3.4. Reconstruction with Independent Component Analysis

In this part, first ICA and fast ICA are briefly introduced, and then, the procedure of using ICA is described.

3.4.1. ICA

Independent component analysis (ICA) can separate or recover unknown independent sources from signal mixture observation (Chen & Khashanah, 2015; Hyvärinen & Oja, 2000). Let T be t[1], ..., t[n] observations with centering and whitening as a pre-processing step. Then, ICA can be written as

$$T = AS, \tag{4}$$

where A is the transformed mixing matrix (Hyvärinen & Oja, 2000) and S is a matrix that contains k independent components of T, where $k \leq n$ and each component is assumed to be non-Gaussian. ICA tries approximate A by maximizing the Independence of S, and the background assumption is that components of S are independent statistically. After acquiring A, the inverse matrix of it W is calculable, and then

$$S = WT. \tag{5}$$

As mentioned, a basic assumption for the ICA is non-Gaussianity. A classical method to measure non-normality is excess kurtosis which is computationally efficient and theoretically straightforward. Nonetheless, kurtosis is sensitive to outliers which makes it non-robust for non-Gaussianity measurements (Hyvärinen & Oja, 2000). Instead, in the literature, the Negentropy method is used which is based on information theory entropy (Hyvärinen & Oja, 2000).

Entropy explains the degree of freedom of information for a given variable. Considering Y as a discrete random variable and, p as probability, then the entropy of H is:

$$H(Y) = -\sum p(Y)\log p(Y). \tag{6}$$

J, the Negentropy, is to obtain non-normality measurement, and it is similar to differential entropy value of which is always non-negative, and for Gaussian variables (i.e., Y_{gauss} of the same Y matrix) is equal to zero. J can be shown as

$$J(Y) = H(Y_{gauss}) - H(Y) \tag{7}$$

In (Hyvärinen & Oja, 2000) an efficient approximation for Negentropy is proposed, which is used in the following FastICA and is written as:

$$J(Y) \propto \{E[G(Y)] - E[G(v)]\}^2, v \sim N(0,1), \tag{8}$$

where $G(x) = \log\cosh(x)$ that is non-linear and non-quadratic.

3.4.2. FastICA

The efficient Negentropy J used to estimate W in (8) is called FastICA. With the assumption of having T centered and whitened, FastICA can estimate W that in (Hyvärinen & Oja, 2000), it is shown W estimations are consistent.

Finally, as in (5) an approximation of unknown sources S is generated, and by remixing with A in (4), an approximation of original time-series signals is achieved, which is used as a reconstruction of original signals. Practical employment is described more in-depth in the section 4.

3.5. Dynamic Time Warping:

Dynamic Time Warping (DTW) is largely used in MTS, such as speech signals as well as classification tasks (Górecki & Łuczak, 2015). In this work, DTW is used to find an optimal global alignment of query and reference signals. It uses a Dynamic Programming (DP) matching approach to generate a distance matrix and pair-wise matching of each sample of the query TS and the nearest sample from the reference signal (Sakoe & Chiba, 1978). For this task, euclidean distance and symmetric step pattern with slope constraint condition of $P = 0$ are applied. DP equation (i.e. $g(i, j)$) for initial point

is:

$$g(1,1) = 2d(1,1). \tag{9}$$

and for the rest is:

$$g(i,j) = \min \begin{bmatrix} g(i,j-1) + d(i,j) \\ g(i-1,j-1) + 2d(i,j) \\ g(i-1,j) + d(i,j) \end{bmatrix}, \tag{10}$$

where d is the distance function (i.e. $d(i,j) = \|a_i - b_i\|$).

3.6. Network Architectures

As mentioned in2, five models are applied and optimized in this work, as shown in Figure 2.

- **MLP**: Multi-layer Perceptron consists of stacks of fully connected dense layers followed by an activation layer to increase non-linearity and prevent saturation of the gradients. Also, a dropout layer is used at the layer's input to prevent overfitting, and the model ends with a softmax layer (Wang et al., 2017). In our experiments, the number of layers, activation function, number of nodes in each dense layer, and dropout rate are considered hyperparameters and get optimized for MLP.
- **FCN**: FCN consists of a few repeated blocks followed by a Global Average Pooling (GAP) layer instead of a fully connected layer, and has a softmax layer as the final layer. Each block starts with a 1-Dimension Convolutional layer followed by a batch normalization layer which speeds up the training process and reduces overfitting, and an activation layer (Wang et al., 2017). In our optimization pipeline, the number of blocks, filter length, the number of filters in 1-D Convolution, and the type of activation function get optimized for this design. In addition, a few optional fully connected layers are added just before the softmax layer, and the number of these layers (can be zero), number of nodes, and dropout rate are to get optimized.
- **ResNet**: ResNet consists of a few residual blocks, and, similar to FCN, ends with GAP and softmax layers. Each residual block has the structure of a 1-D convolutional layer, Batch normalization, and activation layer three times. Also, lateral connections are added from the input of the block to the end of the same block (Wang et al., 2017; He, Zhang, Ren, & Sun, 2016). The number of blocks, filter length, the number of filters in 1-D Convolution, and the type of activation function get optimized for this design. Furthermore, similar to FCN, optional dense layers are added, and their parameters get optimized during architecture optimization.
- **Encoder**: This design is inspired by FCN. However, the GAP layer is replaced with an attention layer (Bahdanau, Cho, & Bengio, 2014) in which the network learns which parts of the time series are more important for classification. The encoder model also consists of a few repeated blocks; each block starts with 1-D convolution followed by instance normalization (Ulyanov, Vedaldi, & Lempitsky, 2016), Parametric ReLU (He, Zhang, Ren, & Sun, 2015) activation, and finally, a dropout. Each block is followed by max-pooling except the last layer, which gets fed into the attention layer, followed by a dense layer and a softmax. For this design, the number of repeated convolutional blocks, number of filters and filter length of 1-D convolutions and drop rate of dropout layer, activation of dense layer, and number of its nodes.
- **XCM**: This design starts with two separate paths. One path uses 2-D convolutions, batch normalization, and activation, and finally uses 1x1 convolution and activation. The second path is similar to path one, except it uses 1-D convolutions. Next, the results of these two paths get concatenated and then pass through one more 1-D convolution followed by GAP and softmax layer. Having two separate paths help to extract feature and time attention of the input using gradient-based methods. For this design, filter length, the number of filters, and activation function type gets optimized.

4. IMPLEMENTATION

Arc Stud Welding is described in (Samardžić et al., 2007; Al-Sahib et al., 2009; Hildebrand & Soltanzadeh, 2014; Ramasamy, Gould, & Workman, 2002), and the same procedure (shown in Figure 3) is performed for experiments in this work, except the process is automated through a robotic arm. During the welding process, sensors in the robotic arm actuator monitor voltage and current. Simultaneously, the lift position sensor also monitors fastener positioning and the reference signal for lift position. LMcurrent is the current of the linear motor that moves the stud and is a critical factor in the penetration of the surface material. For these experiments: The average penetration is 0.70 with an average welding time of 72 ms. The sampling rate is approximately 1 kHz, and finally, all logged sensors' data get transmitted through MQTT protocol to generate a dataset of defective welds.

In the following, the dataset is described, and synthetic signal generations, as well as prediction models, are elaborated on.

4.1. Dataset

The initial dataset contains six classes of Arc Stud Welds with various welding conditions, as mentioned, and measurement types include Voltage, Current, LM Current, and its reference, lift position, and its reference (residual of which is called penetration). Additionally, synthetic measurements, including Power (i.e., pair-wise V*I for each feature) and Resistance (i.e., V/I), are also computed and added to the dataset. Therefore, there are six measurement types in total. Although Current is a controlling parameter of the process, experiments show that it helps network classify more accurately. Hence,

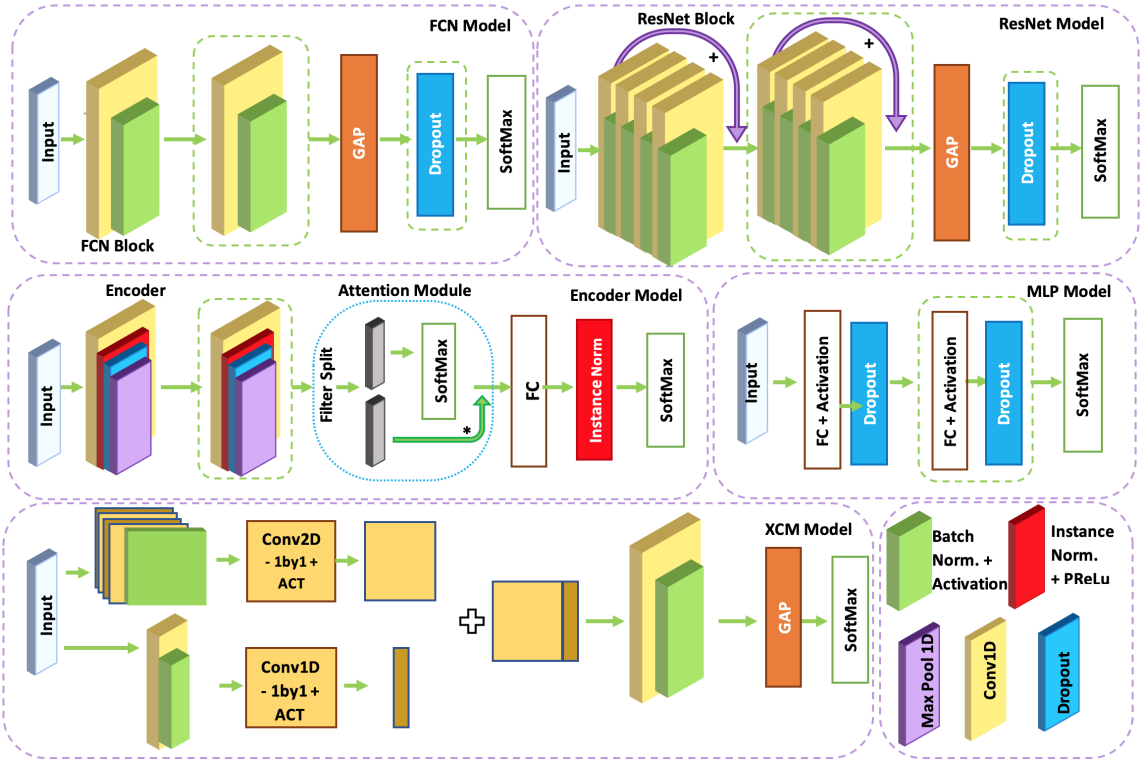

Figure 2. The models' architecture. The blocks inside Lime Dashed lines are repeated blocks, and the number of repeats can from 1 to 3. The optimal repeat number is searched for during optimization.

Classes	*RC1*	*RC2*	*RC3*	*RC4*	*RC5*	*RC6*
Count	59	60	56	52	16	20

Table 1. Class Distribution for Dataset

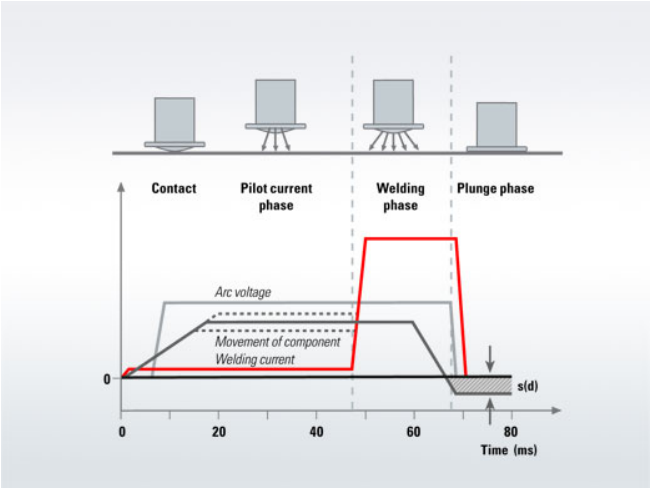

Figure 3. Arc Stud Welding Process (Stanley Black and Decker, 2022).

it is kept among measurements.

As mentioned in this process, defective welds are synthetically generated with known root causes of the defects. The six types of defects (or root causes) are synthetically generated based on the design of the experiment principles. The defect types include four types of pollution on the welding surface which are coded into Root Cause (RC) 1 to Root Cause (RC) 4. Two types of defects are also made on stud quality and electrical system and coded into Root Cause 5 and Root Cause 6. Figure 4 visualizes how measurements for various Root Causes across classes are similar, illustrating how patterns are complicated to extract, and why additional synthetically generated measurements and residual signals are helpful. The final dataset has 263 samples from 6 classes, and the distribution is shown in Table 1. Each instance has six measurements with 480 timestamps. The final dataset (i.e., residue dataset) has six measurements of Voltage residual, Current residual, Penetration, LMCurrent residual, Power, and Resistance.

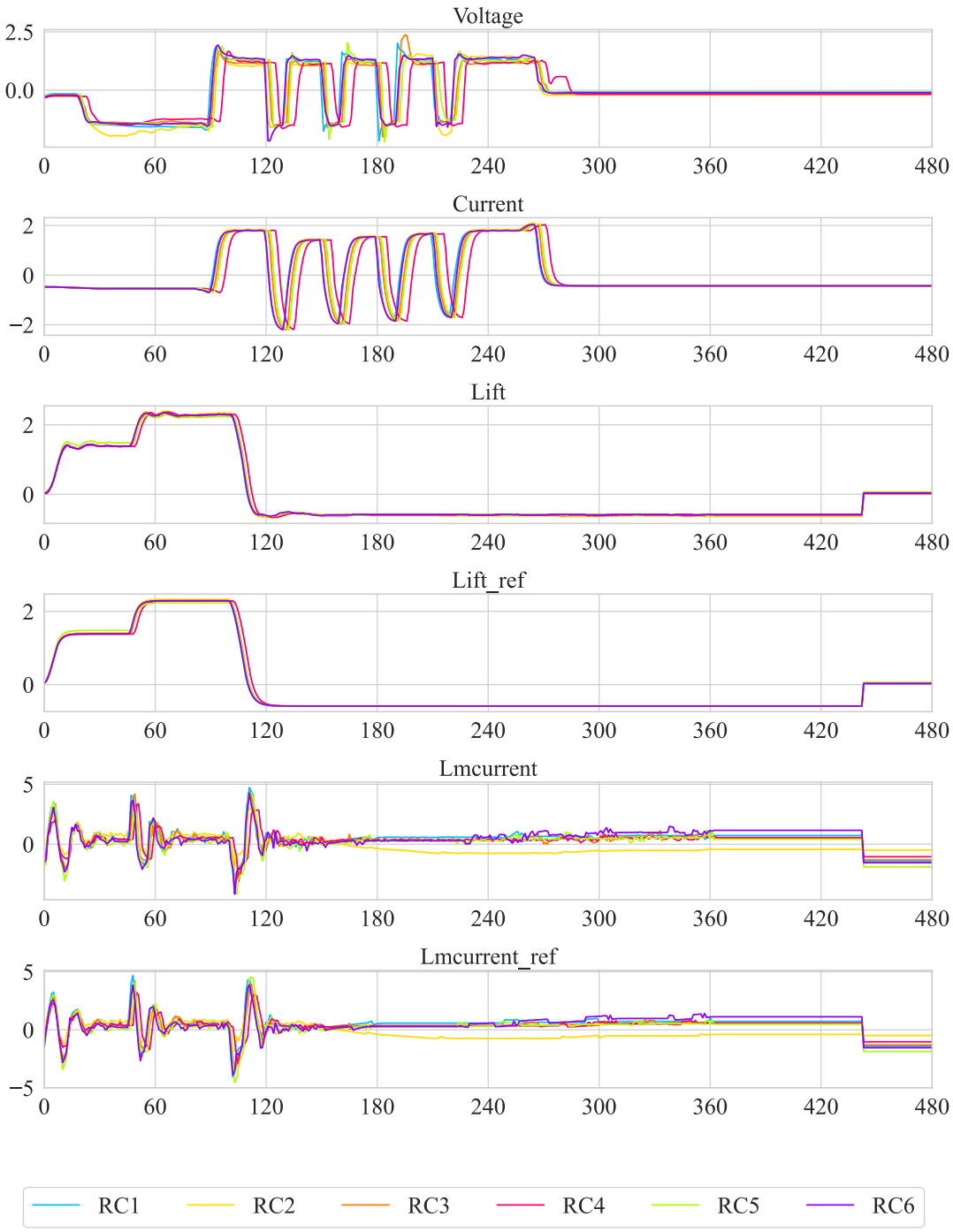

Figure 4. The similarity of measurements across classes RC1 to RC6. The full length of signals is visualized (i.e., 0 to 480).

4.2. Pre-processing

Initially, only voltage and current have a length of 480. Other measurements are padded with zeros to have the same length. For LM current, at the end of the process, it is stopped, and thus zero-padding is meaningful. For the lift position, since the nozzle turns back to the initial point, then padding with zero is rational and meaningful.

4.3. Synthetic Measurement and Residual Generation

As described in 3, a series of steps are performed to generate synthetic residuals to use as a reference for voltage and current. Starting with a mean approximation of signals to get residuals. Since for LM and lift position a reference signal exists, the existing reference will be used as $\hat{T}$ and residual (T') is $T' = T - \hat{T}$. For measurements with no existing reference signal, a mean approximation is required. As mentioned in 3.4, since independent components of unknown sources of S in (5) are an approximation of the original signal T, remixing through (4) creates an approximate of the original signal with common unknown sources. Hence, the resulting signal ($\hat{T}$) is employed for creating a residual signal. Therefore, to acquire mean reconstruction ICA, for measurements with no reference signal (i.e., voltage, power, resistance) and per stud type, a three-component ICA transform and inverse transform (reconstruction) are done, respectively. Then, the mean of reconstructed signals (all signals have the same stud type, measurement type, and features) is calculated and used as $\hat{T}$ to get T' residual. For synthetic data (i.e., power and resistance), multiplication or division is done prior to residual calculation. Finally, FastICA from (Pedregosa et al., 2011) implementation is applied for the mentioned reconstructions.

Next, dynamic time warping is applied to the signals. Since the start and the end of extreme periods or peaks change along with signals, then extreme periods in the query signal and mean signal do not overlap, and this results in shifts in the final residual. Hence, using DTW on residual (T') as the original signal and (T) as a reference would align peaks with the query reference signal. Note that for signals with existing reference signals, DTW is not required, for no shifts appear in those. Moreover, it is experimentally determined that applying symmetric step pattern results in a smaller rooted mean square error in comparison with an asymmetric step pattern. Finally, Algorithm 1 summarizes how the residual calculation is done to create the residue dataset.

As the last step, normalization is done per sensor measurement type using (3). Figure 5 shows samples of normalized voltage and current, its mean reconstruction, and aligned and non-aligned residual signals.

Algorithm 1: residual calculation for a specific sensor measurement type

Result: residuals of T' as residuals_set ;
residuals_set ← [ ] ;
if *reference signal ($\hat{T}$) exists* **then**
  while *not processed every instance T* **do**
    $T' \leftarrow T - \hat{T}$;
    residuals_set ← T' ;
  end
else
  ICA,$\hat{T}_{mica}$ ← [] ;
  ICA ← fit ICA for the entire set of Ts ;
  ICA ← Inverse Transform ICA ;
  $\hat{T}_{mica}$ ← calculate mean of reconstructed signals over x-axis;
  while *not processed every instance T* **do**
    $T \leftarrow DTW(T, \hat{T}_{mica})$;
    $T' \leftarrow T - \hat{T}_{mica}$;
    residuals_set ← T' ;
  end
end
normalize residuals_set using (3);

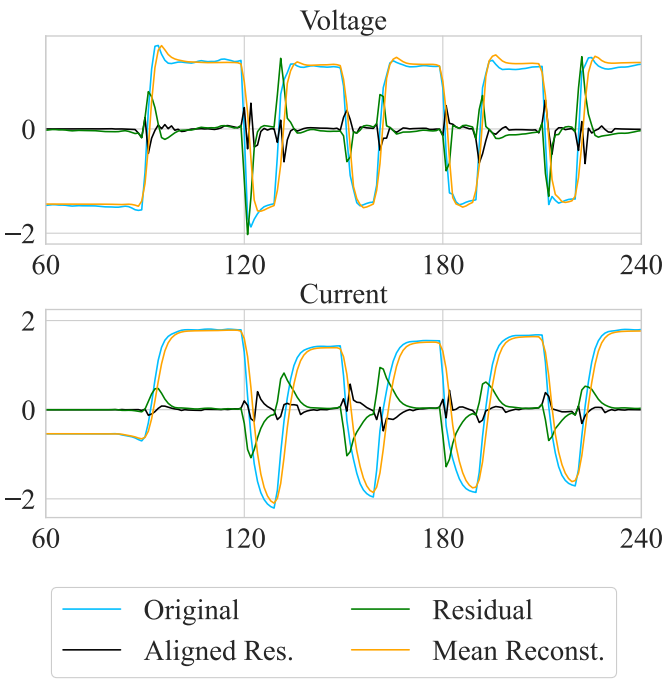

Figure 5. Samples of voltage and current for original normalized measurements, their ICA mean Reconstructions for a specific sample, their residuals, and DTW aligned residuals. Only timestamps 60 to 240 are visualized for a clear demonstration.

Parameters	*Search Space*
optimizer	Adam, SGD, RMSprop, Adadelta
activation functions	relu, swish, sigmoid, tanh, selu
# of layers/blocks	2 to 5
# of filters	16, 32, 64, 128, 256
kernel size	1 to 12 (dependent to model)
drop out rate	0 to 0.5 with 0.1 step
Batch size	32 to 256 with 32 step
Epochs	40 to 100
learning rates	1e-2 to 10e-4 (logarithmic)
max-pooling size	2 & 4 (for Encoder only)
Dense layer nodes	128 & 512 (when applicable)

Table 2. Hyper-parameters search domain used for Bayesian Optimization.

4.4. Classification models

4.4.1. Hyper-parameter Optimization

Residue dataset gets fed into several architectures described in section 3.6, and shown in 2 including MLP, FCN, ResNet, Encoder, and XCM. The models have a pre-defined number of blocks, and layers as described in the original works. To show the effectiveness of Algorithm 1, the models are once trained with default parameters as used in (Fawaz et al., 2019), and then systematically optimized. For optimization, all hyper-parameters and overall architectures are optimized using Bayesian Methods. Table 2, illustrates search space for Bayesian Hyperparameter Optimization (HPO). Max-pooling is only used for the Encoder model, while the rest of the parameters are common for all models.

4.4.2. Training Setting

All models are optimized and tested on a single NVIDIA GTX 1080 GPU. The implementation and HPO are done using Tensorflow v2.3 and Keras v2.1. Keras-Tuner (O'Malley et al., 2019) is used for HPO. Keras-Tuner is a Keras-friendly framework, with implementations of HPO algorithms. It is possible to smoothly manage to optimize activation functions, kernels, layers, and architecture of a descriptive Keras implementation. For the experiments, Bayesian optimization is used with the Gaussian Process model as the surrogate model, and the acquisition function used is the upper confidence bound (UCB). Also, the objective was set to maximize validation set accuracy, with 50 initial points.

Single model training is performed in two to three minutes; However, HPO takes between two to five hours depending on model size, number of parameters, and search space. Finally, inference time is around a millisecond (ms) for a batch size of one and two ms per batch with a size of 32, meaning the model is able to predict in real time.

4.4.3. Optimized Models

The residue dataset contains six measurements, which are padded to have similar lengths. Thus, the models are fed with tuples with shape (batch size (bs), sequence length, number of signals), i.e., (bs, 480, 6). 80% of the samples of the dataset stratifiedly sampled for training and 5-fold cross-validation, and the remaining 20% is withheld for unbiased testing. Early Stopping with the patience of 15 epochs is also applied. For models trained with the raw dataset, the learning rate is fixed to 1e-3 with Adam optimizer, batch size of 32, and relu as activation function, the rest of the parameters can be found in (Fawaz et al., 2019). Table 3 summarizes the best case models for each design.

Table 4 reveals average performance, recall, F1-score for all four models.

Table 5 illustrates the performance of the Encoder model as the best model for each class. As it shows, the hardest root cause to detect is RC4.

4.4.4. Results Explainability

As described, XCM is designed to be capable of providing explainability for multivariate time-series datasets. Figure 6 shows a sample of saliency map for a True positive prediction from class RC2, in which the red areas show high attention to the network, while white and blue show no to little attention. Figure 6-a describes that most of the attention of the model is to the detection of a defect from RC2 is related to the mid-part and ending part of penetration. Nonetheless, the network is not considering current and voltage residual (i.e., current* and voltage*), and the minimum consideration is for Resistance and Power. Figure 6-b relates to time attention, and it is related to the combinatorial path of the network (i.e., the path with 1-D convolutions), which represents that mid-part of the signal is taken into consideration for RC2 defect detection and relates to the second cycle of the welding process.

5. DISCUSSION

5.1. Model Performance

Table 3 shows the best parameters extracted using the Bayesian method. As demonstrated for all models, Adam is the best optimizer, and in most cases, ReLU showed high performance as an activation function. As mentioned in Figure 3.6, optional fully connected layers were added after the GAP layer and based on Table 3, these show noticeable improvement in overall network performance. Moreover, models FCN, Encoder, and ResNet tend to start with smaller window sizes and the number of filters, and these increase further on deeper layers.

Table 4 demonstrates model performance on both test sets and averaged F1-score of 5-fold on the validation set. It reveals

Parameters	***Models***				
	MLP	***FCN***	***ResNet***	***Encoder***	***XCM***
Optimizer	Adam	Adam	Adam	Adam	Adam
activation fun.	tanh	relu	swish	relu	relu
learning rate	10e-4	10e-2	10e-4	10e-4	10e-4
Epochs	100	100	100	100	100
Batch size	32	32	32	32	32
# of blocks	4	3	3	2	2
# maxpooling	-[a]	-	-	2	-
# of filters	-	16|256	32|64|64	256|512	128|128
kernel sizes	-	12|12	6|4|9	5|12	20|20
Dropout	0.4|0.5|0.1	0.2|0.5	-	0.5	-
Dense	350|350|500	128|32	-	128	-
# of param.	1.3M	89K	257K	1.7M	38K

a denotes that the parameter does not exist for the specified model.

Table 3. Hyperparameters extracted for various architectures were extracted using Bayesian Optimization.

	Models	***Test Set***			***Validation Set***
		Precision	***Recall***	***F1-score***	***F1-score***[a]
raw dataset	MLP	0.817	0.807	0.805	0.76
	FCN	0.66	0.596	0.551	0.516
	ResNet	0.59	0.654	0.616	0.52
	Encoder	0.834	0.769	0.767	0.759
	XCM	0.65	0.53	0.52	0.51
residue dataset (our method)	MLP	0.81	0.807	0.799	0.853
	FCN	0.682	0.71	0.668	0.776
	ResNet	0.73	0.71	0.71	0.75
	Encoder	0.847	0.847	**0.843**	0.849
	XCM	0.59	0.57	0.58	0.56

a These values are averaged F1-score of validation over 5-fold.

Table 4. Models performance on test/validation sets. F1-score is weighted to address class imbalance on final accuracy.

Class	***Test Set***		
	Precision	***Recall***	***F1-score***
RC1	0.85	1.0	0.92
RC2	0.70	0.70	0.70
RC3	0.73	0.67	0.70
RC4	1.0	0.67	0.80
RC5	1.0	1.0	1.0
RC6	1.0	1.0	1.0
Weighted Average	**0.85**	**0.85**	**0.84**

Table 5. metrics per class for best model (Encoder).

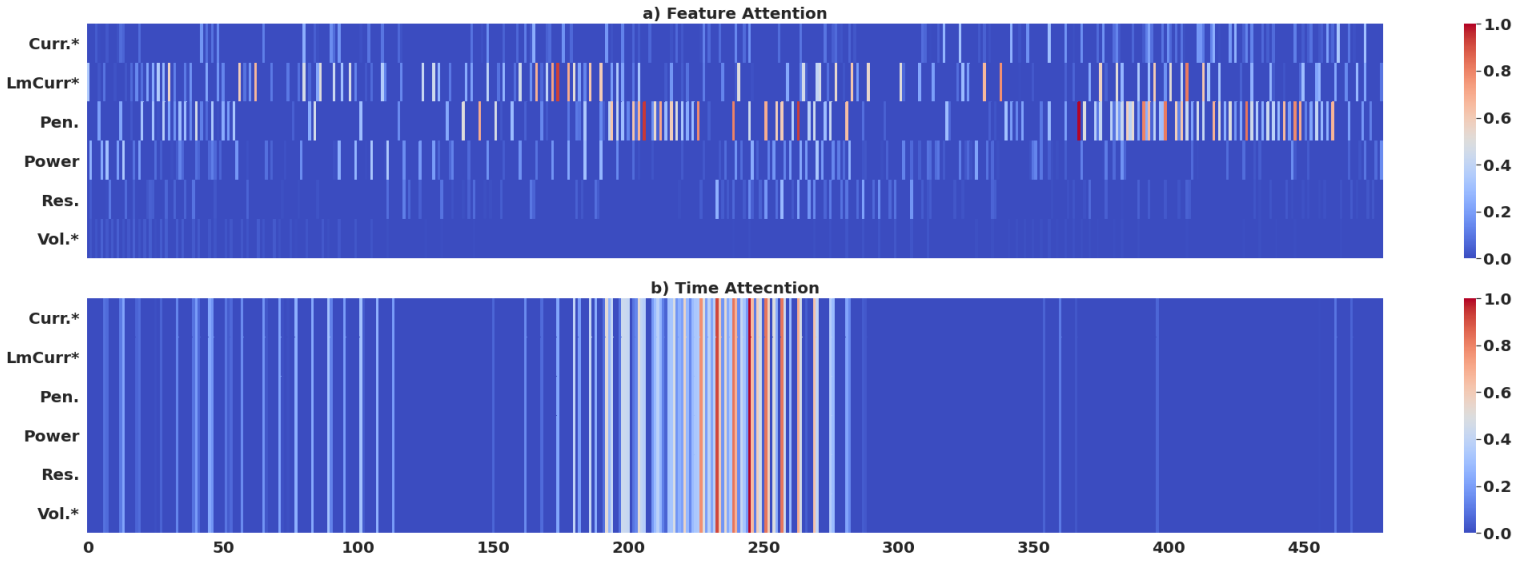

Figure 6. Feature Attention-a, Time Attention-b for prediction of class RC2. the x-axis is timestamps and the y-axis is features, Current*, LmCurrent, Penetration, Power, Resistance, Voltage*. All are residual signals. Penetration is the difference between lift position and its reference, * underlines that residual is used.

that models are showing great performance on about 65% of training data, and the best model is the Encoder model. Table 5 shows per-class accuracy for all classes using Encoder Model. Classes related to RC5 and RC6 are the easiest to classify, while RC2, RC3, and RC4 are the hardest. Although the Encoder model shows a 0.70+ F1 score per class, the same does not happen with the per-class F1 score of other models. For instance, FCN shows a 0.85+ F1-score for classes RC2 and RC3, while it fails to detect RC4 class (i.e., less than 0.4 accuracy). Thus, a model ensemble will help to reach higher overall performance.

Class	*Measurement Attentions*
RC1	Vol*, LMCurr*, Curr*, Res.
RC2	Vol*, LMcurr*, Curr*
RC3, RC4	-
RC5, RC6	Pen., Res.

Table 6. Per Class XCM Measurement Attentions

5.2. Model Explainability

XCM model feature attentions on the test set reveal feature importance in for each classification. Figure 6 demonstrates an example for RC2 class, which is a workpiece defect. The overall analysis of feature importance is summarized in Table 6, where features importance for each class is sorted in order of their contribution (i.e., left most feature is the one with the most contribution. Linear Motor current is the feature with the most contribution in detection. One explanation is, the impurity on the surface causes slight movements when the welding head makes contact with the surface, which results in variations in LMCurrent. These slight movements vary across classes, and are a reliable source for the model to make classification decisions. On the other hand, for stud-related impurities (i.e., RC5, RC6), the common discriminative factor is Penetration. Although the rest of the signals had partial contributions to classification, (illustrated in Figure 6), their existence is crucial to achieving the highest performance. A witness to this is the results in Table 4, where models without synthetic measurements have lower accuracy. Finally, since the XCM model performance is not acceptable for RC3, and RC4, they are not reported.

5.3. Welding Improvements

The described procedure can appropriately and immediately detect issues related to the base material, stud, workpiece, or configuration, where slight degradation can cause unexpected issues. As the system and monitoring are standalone in the process, it does not have any interference with the welding operation. Moreover, this allows overtime monitoring of the system and components, where unexpected changes and downtime can be monitored and addressed during scheduled maintenance rather than causing unexpected emergency downtime, which increases productivity and prolongs the system's life span. Furthermore, adaptive weld parameter control is achievable based on surface impurity. For example, based on observed primer on the surface weld time should be optimized to increase or decrease energy transfer, and reduce surface errors.

6. CONCLUSIONS

In this paper, an initial dataset of Stud Welding is generated and transformed into the residue dataset with a novel method to boost classification performance. Moreover, Bayesian hyper-parameter optimization and comparative analysis of deep convolutional classifiers for MTS are conducted. Furthermore, the explainability of the model, feature, and time attention based on XCM architecture is analyzed. Our experiments show the combination of our approach of synthetic measurements, residual signal generation, and deep classifier has enabled further investigations on tracking possible causes of stud weld failure. The residue dataset creation steps consist of the mean approximation of signal through ICA reconstruction, which is then used as a reference for calculating signal residual and is followed by performing dynamic time warping for signal alignment. For classifiers, MLP, FCN, ResNet, Encoder, and XCM are investigated and optimized both in terms of architecture and parameters. The most accurate model is the Encoder model, with an F1-score of 0.84 on both the 5-fold cross-validation and test set.

Although the approach reveals high-performance defect classification, further investigation is required for both improving existing designs and production deployment. For improving present designs, ensemble methods, extending the dataset, and investigation feature importance are suggested. Another interesting path can be a deep investigation of the explainability of models. XCM is chosen as it outperforms other existing models and utilizes ten times fewer parameters than MTEX-CNN. However, the explainability capabilities are still to be investigated. For production requirement, investigations for validating the same performance is suggested as production is not labeled, and hand labeling such data is either impossible in most cases or time-consuming and costly.

REFERENCES

Al-Sahib, N. K. A., Ameer, H. K. A., & Ibrahim, S. G. F. (2009). Monitoring and quality control of stud welding. *Al-Khwarizmi Engineering Journal*, *5*(1), 53–70.

Assaf, R., Giurgiu, I., Bagehorn, F., & Schumann, A. (2019). Mtex-cnn: Multivariate time series explanations for predictions with convolutional neural networks. In *2019 IEEE International Conference on Data Mining (ICDM)* (p. 952-957). doi: 10.1109/ICDM.2019.00106

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Baydogan, M. G., Runger, G., & Tuv, E. (2013). A bag-of-features framework to classify time series. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(11), 2796–2802.

Chambers, H. A. (2001). Principles and practices of stud welding. *Pci Journal*, *46*(5), 46–59.

Chen, K.-H., & Khashanah, K. (2015). The reconstruction of financial signals using fast ica for systemic risk. In *2015 IEEE Symposium Series on Computational Intelligence* (p. 885-889). doi: 10.1109/SSCI.2015.130

Dong, J., Xu, G., Yu, H., Fan, G., Wei, L., & Gu, X. (2019). Connection status evaluation in arc stud weld joints by ultrasonic detection. *The International Journal of Advanced Manufacturing Technology*, *100*(1), 663–672.

Ducoffe, M., Haloui, I., & Gupta, J. S. (2019). Anomaly detection on time series with wasserstein gan applied to phm. *International Journal of Prognostics and Health Management*, *10*(4).

Fauvel, K., Lin, T., Masson, V., Fromont, É., & Termier, A. (2020). Xcm: An explainable convolutional neural network for multivariate time series classification. *arXiv preprint arXiv:2009.04796*.

Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, *33*(4), 917–963.

Goldin, D. Q., & Kanellakis, P. C. (1995). On similarity queries for time-series data: constraint specification and implementation. In *International Conference on Principles and Practice of Constraint Programming* (pp. 137–153).

Górecki, T., & Łuczak, M. (2015). Multivariate time series classification with parametric derivative dynamic time warping. *Expert Systems with Applications*, *42*(5), 2305-2312. Retrieved from https://www.sciencedirect.com/science/article/pii/S0957417414006927 doi: https://doi.org/10.1016/j.eswa.2014.11.007

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1026–1034).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).

Heidarydashtarjandi, R., Prasad-Rao, J., & Groth, K. M. (2022). Optimal maintenance policy for corroded oil and gas pipelines using markov decision processes. *International Journal of Prognostics and Health Management*, *13*(1).

Hildebrand, J., & Soltanzadeh, H. (2014). A review on assessment of fatigue strength in welded studs. *International Journal of Steel Structures*, *14*(2), 421–438.

Hyvärinen, A., & Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural Networks*, *13*(4-5), 411–430.

Keogh, E., & Ratanamahatana, C. A. (2005). Exact indexing of dynamic time warping. *Knowledge and Information*

Systems, *7*(3), 358–386.

Naddaf-Sh, M.-M., Naddaf-Sh, S., Zargarzadeh, H., Zahiri, S. M., Dalton, M., Elpers, G., & Kashani, A. R. (2021). Defect detection and classification in welding using deep learning and digital radiography. In *Fault Diagnosis and Prognosis Techniques for Complex Engineering Systems* (pp. 327–352). Elsevier.

Naddaf-Sh, S., Naddaf-Sh, M.-M., Zargarzadeh, H., Dalton, M., Ramezani, S., Elpers, G., … Kashani, A. R. (2022). Real-time explainable multiclass object detection for quality assessment in 2-dimensional radiography images. *Complexity*, *2022*.

O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L., et al. (2019). *Kerastuner.* https://github.com/keras-team/keras-tuner.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others (2011). Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, *12*, 2825–2830.

Ramasamy, S., Gould, J., & Workman, D. (2002). Design-of-experiments study to examine the effect of polarity on stud welding. *Welding Journal New York*, *81*(2), 19–S.

Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *26*(1), 43-49. doi: 10.1109/TASSP.1978.1163055

Samardžić, I., Klarić, Š., & Siewert, T. (2007). Analysis of welding parameters distribution in stud arc welding. *Welding & Materials Technical, Economic and Ecological Aspects*, 791–800.

Schäfer, P. (2015). The boss is concerned with time series classification in the presence of noise. *Data Mining and Knowledge Discovery*, *29*(6), 1505–1530.

Serrà, J., Pascual, S., & Karatzoglou, A. (2018). Towards a universal neural network encoder for time series. In *CCIA* (pp. 120–129).

Stanley Black and Decker. (2022). *Stud welding products systems.* Retrieved from http://www.emhart.eu/eu-en/products-services/products-by-category/tucker-stud-welding/stud-welding-systems/energy-units.php (Accessed: 04-26-22)

Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2016). Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*.

Wang, Z., Yan, W., & Oates, T. (2017). Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International Joint Conference on Neural Networks (IJCNN)* (pp. 1578–1585).

Zhang, W., Jha, D. K., Laftchiev, E., & Nikovski, D. (2019). Multi-label prediction in time series data using deep neural networks. *International Journal of Prognostics and Health Management*, *10*(4).