

PAPER • OPEN ACCESS

## A simple machine learning based framework for processing the inline inspection data of subsea pipelines

To cite this article: Z Liu *et al* 2021 *IOP Conf. Ser.: Mater. Sci. Eng.* **1201** 012050

View the [article online](#) for updates and enhancements.

### You may also like

- [Inline ECE measurements for NTM control on ASDEX Upgrade](#)  
H. van den Brand, W.A. Bongers, J.K. Stober *et al.*
- [Estimation of carbon storage and carbon density of forest vegetation in Ili River Valley, Xinjiang](#)  
Guo jing, Zhang renping, Wang ranghui *et al.*
- [Effects of Graphene Oxide and Reduced Graphene Oxide Interlayer Interactions on the Charge Storage Mechanism](#)  
A. J. Saleh Ahammad, Tamanna Islam, Md. Mahedi Hasan *et al.*



The Electrochemical Society  
Advancing solid state & electrochemical science & technology

## 241st ECS Meeting

May 29 – June 2, 2022 Vancouver • BC • Canada

Abstract submission deadline: Dec 3, 2021

Connect. Engage. Champion. Empower. Accelerate.  
**We move science forward**



**Submit your abstract**



# A simple machine learning based framework for processing the inline inspection data of subsea pipelines

Z Liu<sup>1,\*</sup>, S S Dahl<sup>2</sup>, E S Larsen<sup>2</sup> and Z Yang<sup>2</sup>

<sup>1</sup> Western Norway University of Applied Science, Norway

<sup>2</sup> Norwegian University of Science and Technology, Norway

\* Contact author: zhli@hvl.no

**Abstract.** This paper presents a simple machine learning based framework for diagnosing the inline inspection data (ILI) of subsea pipelines. ILI data are obtained by intelligent pigging devices operating along subsea pipelines. The wall thickness (WT) and standoff distance (SO) are collected by the sensors installed on the pigging, which are normally in the format of 2D arrays. There are many uncertainties for the ILI data collected from the offshore survey. An attempt was made to apply the machine learning method to diagnose the uncertainties. A convolutional neural network (CNN) is used, the ILI data are discretized and processed in 64x64 grid size. Fabricated training datasets were made for training the machine learning model since the ground truth information (actual corroded wall thickness) is hardly known in this case. The trained model was successfully. It is demonstrated that certain corrosion patterns have been recognized by the trained model. Comparisons were performed between the new method and traditional methods with case studies on real ILI data. The validity of the methodology was discussed.

## 1. Introduction

According to the Norwegian petroleum directorate (2021)[1], the total length of the Norwegian pipeline network is over 15, 000 kilometers (oil and gas). The integrity management of these pipelines is becoming increasingly important because many of them are nearing the end of their designed lifetime. To have a control of the internal or external corrosions, the inline inspection data (ILI) data obtained by smart pigging are often used. The inspection tool travels along the pipeline, collecting the remaining wall thickness data (WT) and the standing-off data (SO).

Various non-destructive techniques could be used for the sensors, such as Magnetic Flux Leakage (MFL), Ultrasonics Testing (UT), electromagnetic acoustic technology (EMAT) and Eddy Current (EC), as discussed in details by Ma et al. (2021) [2] and Klann and Beuker (2006) [3]. For ultrasonics testing inspection tool (UT), the measuring principle is based on perpendicular incidence of ultrasound into the pipe wall. The ultrasonic pulse is reflected from the back wall and travels several times to and from, until the signal energy is dissipated. The time  $t$  between entry echo and first rear wall echo or between two rear wall echoes is measured. The wall thickness can then be determined by the time and sound velocity. In addition, the distance between probe and pipe wall is measured (stand-off), see Figure 1 for an illustration of the inspection principle. The data for WT and SO are stored as .csv files, which are both 2-dimension arrays, corresponding to the axial and circumferential directions, see Figure 2 for



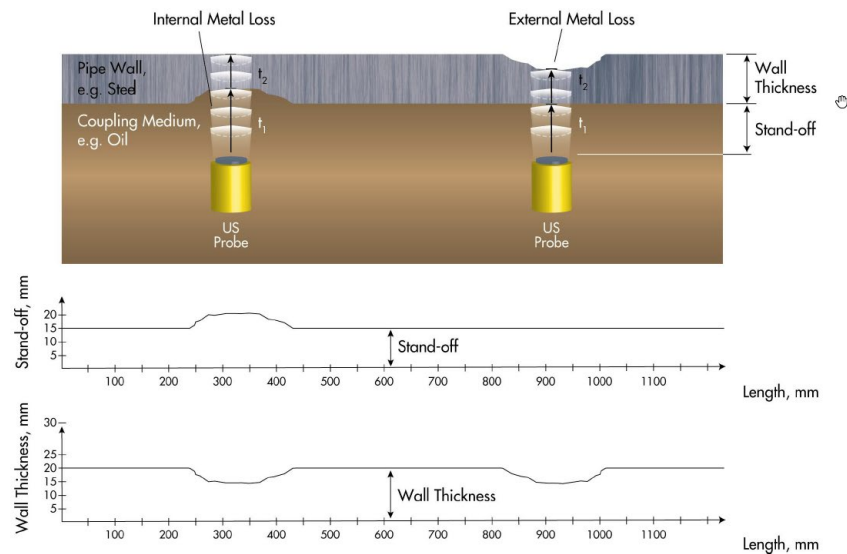
an example of ILI data. The resolutions could be 3.0 mm and 8.0 mm respectively. A large amount of data can be expected for subsea pipelines in several kilometers.

The collected ILI data are normally with certain variations, which could be due to the wall thickness variations, vibrations/radial movement of the sensor, residual debris and erroneous or missing data points. The contractor conducting the survey is expected to provide high quality ILI data, for example, following the principles set out in API 1163 (2013) [4]. However, the raw ILI data are still not suitable for the evaluation of the pipeline's conditions, such as the remaining wall thickness (RWT), the corrosion rate and the remaining bursting capacities. Additional filtering of the raw WT and SO data is required prior to the generation of the so-called River Bottom Profile (RBP), see DNVGL-RP-F101 (2019) [5]. The RBP maps minimum thickness values in the circumferential direction and is used to calculate the remaining pipeline capacity. Henning et al. (2007) [6] compared different methods of generating the RBP profile, such as the classic RBP, SO RBP, Classic and Depth, Median (RBP) and the generalized RBP approach. It was concluded that the maximum depth and the generalized RBP approaches yield more plausible results. Liu et al. (2018) [7] did a comparison between the filtering method provided by DNV GL-RP-F101 (2019) [5] and two other methods, namely the Median and Gaussian filters. The filtering method from DNV GL is similar as the generalized RBP approach from Henning et al. (2007) [6]. It was found that the DNV GL method is more reliable while the median and Gaussian filters provide upper and lower bound curves respectively in most cases. Once RBP file is generated, a moving average file (MAV) could be obtained directly by following the method described in DNV GL-RP-F101(2019) [5] (see discussion in Sect. 2.3 in this paper and Figure 9 for details). The MAV file is used for calculating the remaining pipeline capacities by empirical or advanced numerical FEM simulations.

The methods mentioned above for generating the RBP file are all based on explicit algorithms with certain assumption on dealing with data uncertainties. The machine learning method has been improved significantly in recent years. It has been successfully applied to various scenarios, in which explicit rule-based methods may have limited abilities, such as object recognition, language translation, medical diagnosis, stock market trading, email spam and malware filtering, self-driving cars et al. It is a natural application of machine learning on data processing. This paper presents a simple supervised machine learning model. It is applied to fabricated datasets and promising results have been obtained. The MAV file is used as a performance check for the ML method in this study since it is directly used to determine the remaining capacity of the pipeline, see DNV GL-RP-F101 (2019) [5]. Comparisons have been made with respect to other traditional algorithms. Conclusions and discussions on the validity of machine learning are made accordingly.

## 2. Methodology

As the first step to obtain a machine learning model, sufficient training data shall be provided. The training set with data pairs should contain both inputs and outputs for a supervised machine learning model. The challenge to this study is that the true condition of the pipeline remains unknown even though a large amount inspection data is available. To cope with this, artificially fabricated datasets are used for training. The fabricated datasets are designed to capture various corrosion patterns, containing WT, SO and Y data. The Y data are target outputs for the machine learning model. Randomly generated noises are included for WT and SO data. The generated datasets are formatted according to real ILI data formats (see Figure 2) by Python scripts (version 3.7.4, Van Rossum and Drake, 2009 [9]). Keras (version 2.4.3, Chollet et al., 2015 [10]), running on a TensorFlow platform (version 2.3.0, Abadi et al., 2016 [11]), is chosen as the main tool for training the machine learning model in this study.



**Figure 1.** Ultrasound principle for metal loss inspection and quantitative wall thickness measurement, Barbian et al. (2010) [8].

Circumferential direction (256)

	0	1.4	2.8	4.2	5.6	7	8.4	9.8	...	350.2	351.6	353	354.4	355.8	357.2	358.6
0	21.54	21.54	21.54	21.54	21.54	21.54	21.54	21.54	...	21.54	21.54	21.54	20.68	21.54	21.54	21.54
1	21.54	21.54	21.54	21.54	21.54	21.54	21.54	21.54	...	21.54	21.54	21.54	21.54	21.54	21.54	21.54
2	21.54	21.54	21.54	21.54	21.54	24.57	21.54	21.54	...	21.54	21.54	21.54	21.54	24.63	18.43	21.54
3	21.54	21.54	21.54	21.54	21.54	21.54	21.54	21.54	...	21.54	21.66	21.54	18.94	21.54	21.54	21.54
4	21.54	21.54	21.54	21.54	22.19	21.03	21.54	21.54	...	21.54	21.54	21.54	21.54	21.54	21.54	21.54
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	21.54	21.54	21.54	21.54	21.54	21.54	21.54	21.54	...	21.54	21.54	21.54	21.54	21.54	21.54	23.43
996	21.54	21.54	21.54	21.54	21.54	21.54	21.54	22.06	...	21.54	21.9	21.54	21.54	21.54	18.37	21.54
997	21.54	21.54	21.54	24.9	21.54	21.54	21.54	21.54	...	21.54	21.54	21.54	21.54	22.8	21.54	21.54
998	21.54	21.54	21.54	17.84	21.54	21.54	21.54	21.54	...	21.54	21.54	21.54	21.54	21.54	21.75	21.54
999	18.87	21.54	21.54	21.54	21.54	21.54	20.72	21.54	...	21.54	21.54	21.54	21.54	23.3	21.54	21.54

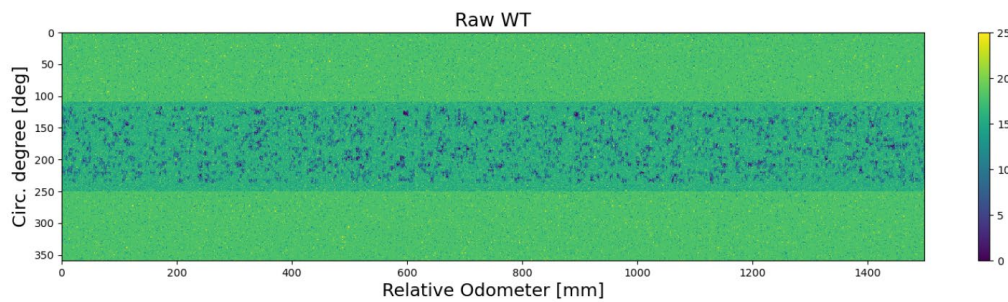
Odometer direction (1000)

**Figure 2.** Example ILI data, a sample data with size of 256 x1000.

2.1. Training data

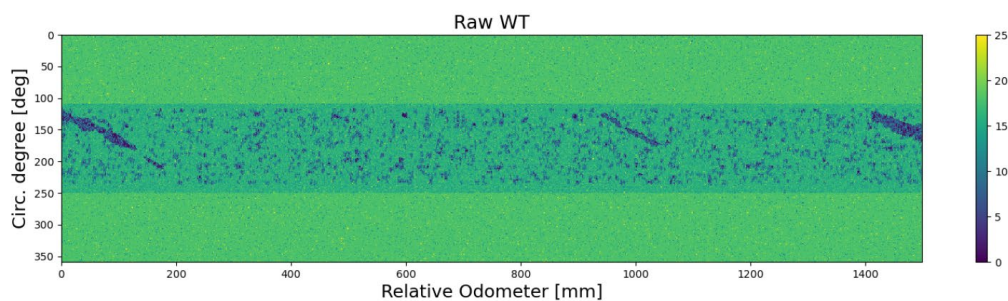
In this study, 25 pairs of training samples are generated, in which 20 pairs are used for training while 5 pairs are used for validation. One training sample consists of two files. The WT file simulates wall thickness data, while the Y file tells the corresponding ground truth of wall thickness. The neural network is then trained with the WT and Y file as input-output pairs. One pipeline section is simulated by creating a 2D matrix with a given dimension, 256 x 1000, which is corresponding to data resolutions on circumferential and axial directions of ILI data respectively (see Figure 2).

- *Channelling corrosion:* Both the WT and Y-section start with the same wall thickness and are identical initially. The combination of uniform and channelling corrosion is simulated on the bottom half of the pipeline. The width of the corrosion area varies. Simulation of the uniform corrosion lowers the wall thickness in the relevant area with a random depth between 0 and 3 mm, see Figure 3 for an example.



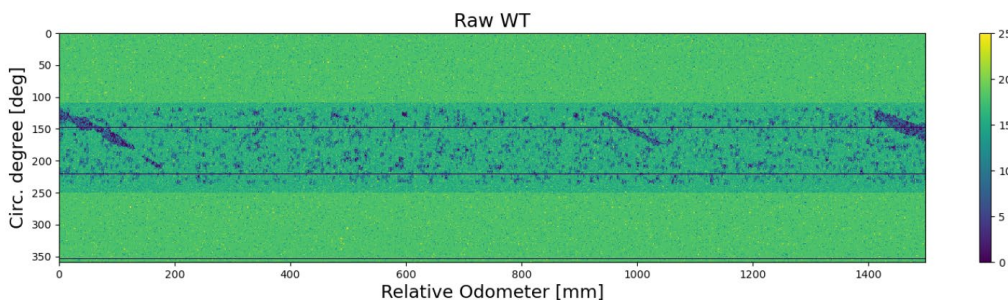
**Figure 3.** Example of simulated pipe section with noise and channelling corrosion.

- *Pitting corrosion*: Pitting corrosion is simulated by stochastically creating pits with different shapes and depth within certain random boundaries, see Figure 4.



**Figure 4.** Example of simulated pipe section with noise, channelling, and pitting corrosion.

- *Missing reading*: Malfunctioning sensors can cause bad readings, resulting in rows or columns with the same WT or missing data reported throughout the section. A random number of zero readings are added to the generated WT data, as shown in Figure 5.



**Figure 5.** Example of simulated pipe section with noise, channelling, pitting corrosion, and missing values.

## 2.2. ML implementation

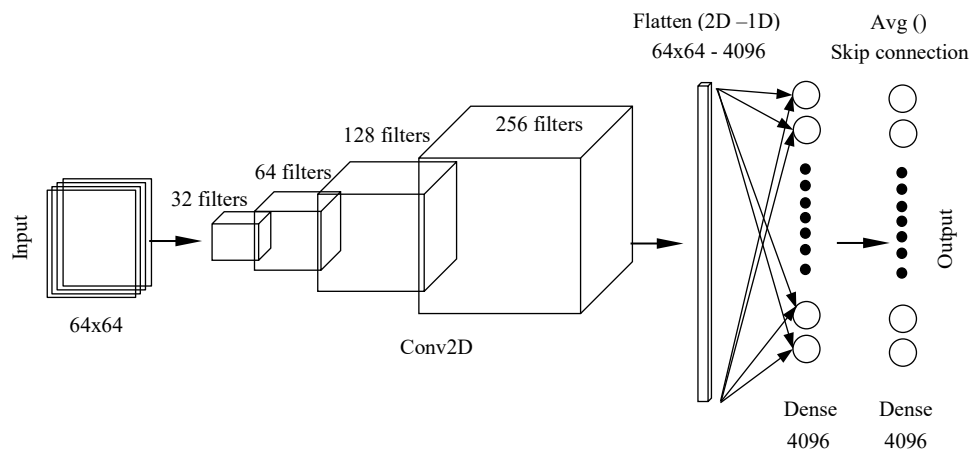
The neural network itself is designed to predict RWT values for a grid size of 64 x 64. Multiple iterations are needed to get satisfying results. The convolutional neural networks (CNN) have proven to be an efficient method after tests. The final architecture ended up being a CNN with a single skip connection.

- *Network architecture*: A CNN that processes the original input image piece by piece. The architecture used is outlined in Table 1. The first layer of the network specifies the input dimensions to the network, which is a 64 x 64 grid. The first layer is implicit in the network and does not alter the input. The following layers are four convolutional layers (Conv2D in Keras for 2D input and output, see Chollet F. et al., 2015 [10]) with an increasing number of filters, starting from 64 to 256. The only difference between the convolutional layers is the number of filters. Apart from the number of filters, the Conv2D layers use ReLU as the activation function,

a kernel size, of 3 x 3, a 2 x 2 stride, and a padding of same. In Keras, a padding value of the same means that a zero padding of appropriate size is added around the input to ensure that the output is about the same size as input after convolution. It is found that the network performed much better with small kernel size (3 x 3) than larger ones (5 x 5 or 7 x 7). The output is flattened and sent into a dense layer after the convolution layers, as seen in Figure 6. The initial input to the network and the output of this dense layer are the sent combined with a skip connection that calculates the average value. Adding this skip connection improved the performance significantly. The last layer is another dense layer that produces an output of shape (4096, 1). This output is then reshaped to the original shape, (64, 64, 1), in a way that correctly recreates the image.

**Table 1.** Each layer of the architecture and its respective configuration.

Layer	Configuration
Input_layer	shape = (64,64,1)
Conv2D_1	filters = 32
Conv2D_2	filters = 64
Conv2D_3	filters = 128
Conv2D_4	filters = 256
Flatten	Input = (64,64,1), output= (4096,1)
Dense_1	Input = (4096,1), output = (4096,1)
Average	Input = (Input_layer, Dense_1), output = (4096,1)
Dense_2	Input = (4096,1), output = (4096,1)

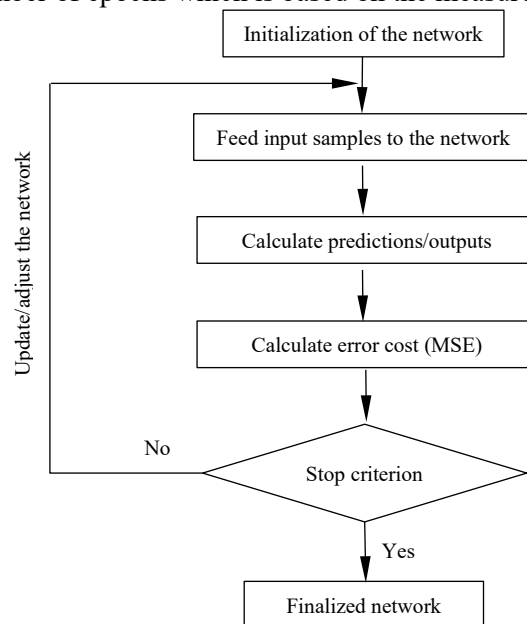


**Figure 6.** Illustration of the finalized network architecture.

- Network configurations:* According to Thakkar et al. (2018) [12], a batch normalization layer is added after every Conv2D layer to increase the learning performance. It is observed that the batch normalization and an increased number of epochs have improved the networks, even for some of the weaker ones. All layers in the network use the default initialization scheme for Keras layers, called the Glorot uniform initializer, see Chollet F. et al. (2015) [10]. This initializer randomly assigns a starting value, within certain limits, to the weights of the network. Consequently, some weights need more significant adjustments than others. The randomness involved is also responsible for slightly different results every time the same neural network architecture is trained. Some early developed networks were trained multiple times from scratch to test the effect of this randomness in this study. The network converged at the same values, with some minor irrelevant differences.

The model is implemented with an initial learning rate of 0.01, which is a relatively high learning rate. The learning rate is gradually reduced during the training of the network to facilitate the fine-tuning of the network. An implemented learning rate scheduler decreases the model's learning rate after every training epoch. The learning rate (LR) can be decreased until a threshold is reached. The loss function used to train the network is the mean squared error (MSE) and the Adam optimizer is used.

- *Training procedure:* A flowchart of the training process is shown in Figure 7. The simulated training samples are fed into the network, the generated outputs are compared to the corresponding targets. If the specified stop criterion has not yet been reached, adjustments are made to the network, which is controlled by the chosen optimizer. The stop criterion used in this study is the number of epochs which is based on the measurement of the performance.



**Figure 7.** Flowchart of the training process.

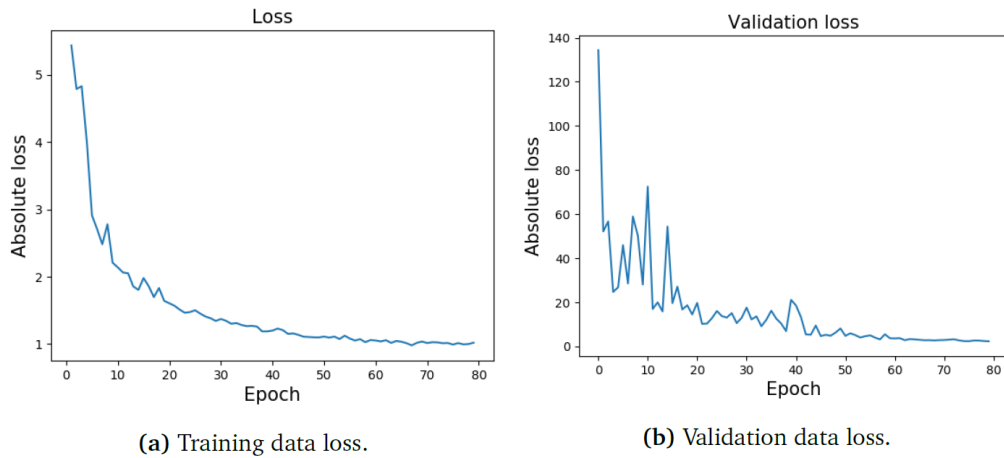
- *Training results:* Figure 8 shows the training loss as well as the validation loss. After 80 epochs, both loss functions have reached convergence.

### 2.3. Performance check

As mentioned earlier, the ground truth value of the remaining wall thickness (RWT) of pipeline inspected is unknown, it is then not possible to quantify the performance of the ML method. Nevertheless, comparisons have been made between the ML method and the traditional DNV GL method based on the MAV file generated, since it is the file that used for calculating the remaining capacity of the pipeline. It is believed that the true values (ground truth) shall be well within the two methods and engineering judgments may be needed to for the final assessment of the pipeline integrity.

The river bottom profile (RBP) files are generated from the remaining wall thickness (RWT) data after filter methods applied. The RBP is created by looping through each sensor around the circumference at each odometer point in the longitudinal direction. The minimum wall thickness is reported.

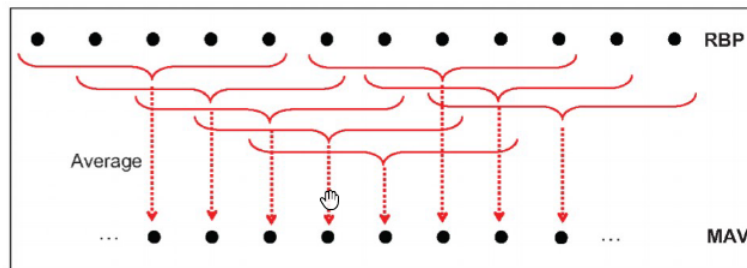




**Figure 8.** Statistics from the training of the ML model.

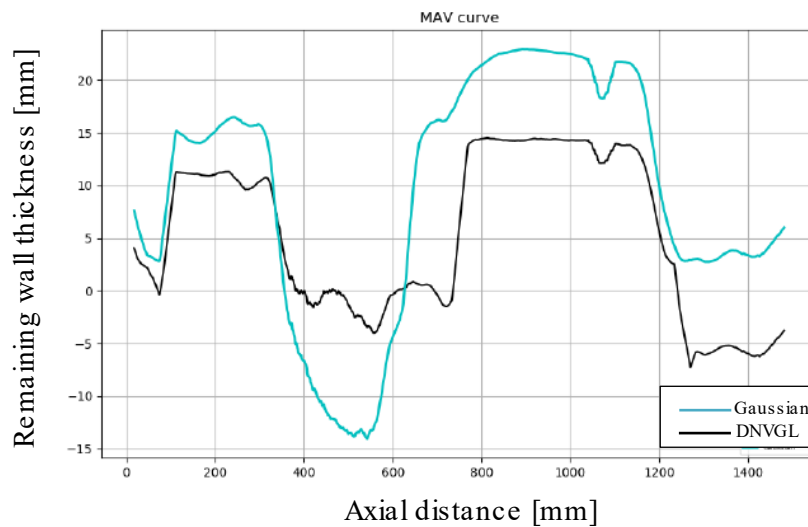
The RBP file is still not suitable for calculating the remaining capacity of the pipeline (the burst pressure). A further smoothing is done by using the moving average method. A moving average profile (MAV) file is then finally obtained. The smoothing is done using the average over a length of  $l=0.5*\sqrt{D*t}$ , where  $D$  is the nominal outer diameter and  $t$  is the original wall thickness as shown in Figure 9.

Figure 10 shows an example of generated MAV file. The MAV file is chosen as a kind of performance measurement for the ML methods in this study. Note that negative values are shown in Figure 10, it is probably due to low quality of data.



**Figure 9.** Illustration of the calculation of moving average from RBP, from DNV GL-RP-F101 (2019).



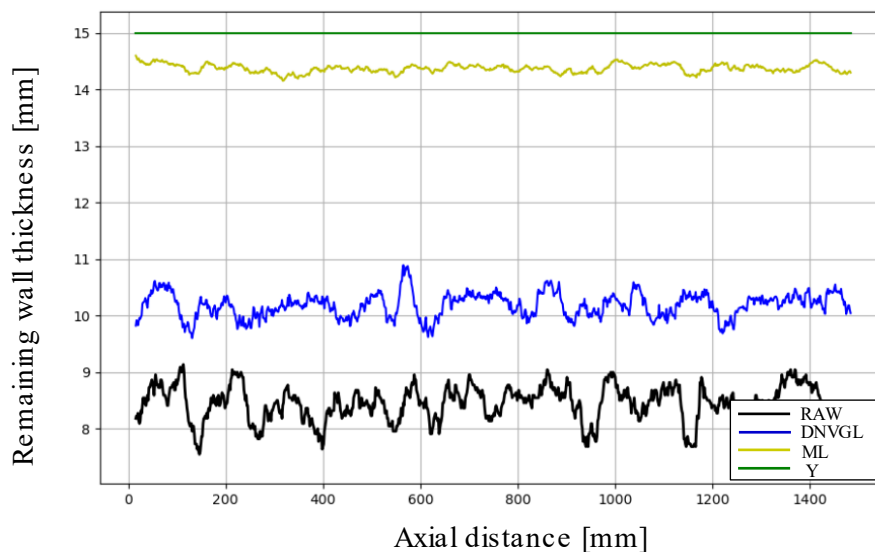


**Figure 10.** Example of MAV file obtained with two rule-based methods.

### 3. Results and discussions

#### 3.1. Simulated data - noise

A simulated data with highly stochastic and spread throughout the whole section. A pipeline section exclusively containing noise is used to isolate the filter's ability to handle noise. The predefined wall thickness is 15 mm before the noise is added. The resulting MAV graphs from DNV GL and ML filter together with the raw WT data and the ground truth Y-data are plotted in **Figure 11**. It is seen that the ML filter gives an RWT close to 14.5 mm, with less variation over the length of the section. The simulated data are simulated by the same algorithm that the training data for the model used in the ML filter. It is seen that the DNV GL method is giving too conservative results in this case as it keeps only the minimum wall thickness along the circumferential direction at each odometer location.



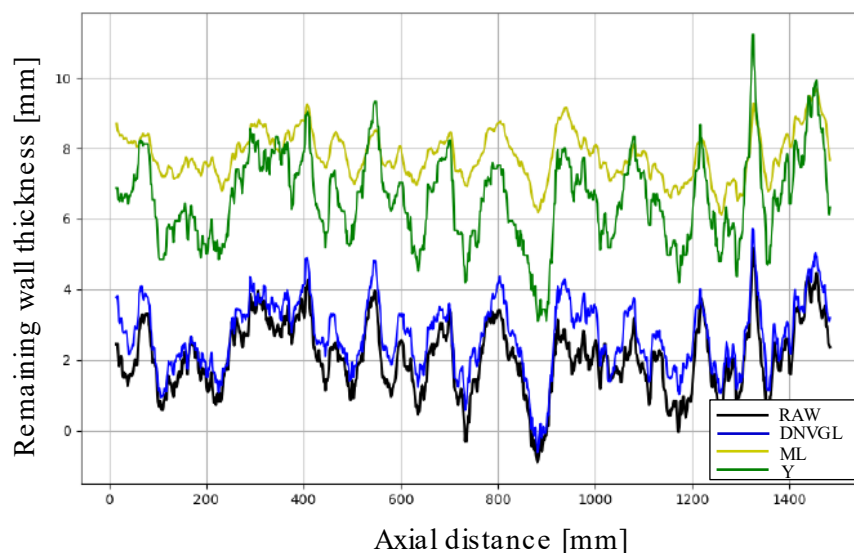
**Figure 11.** MAV generated by different methods.

### 3.2. Simulated channelling corrosion

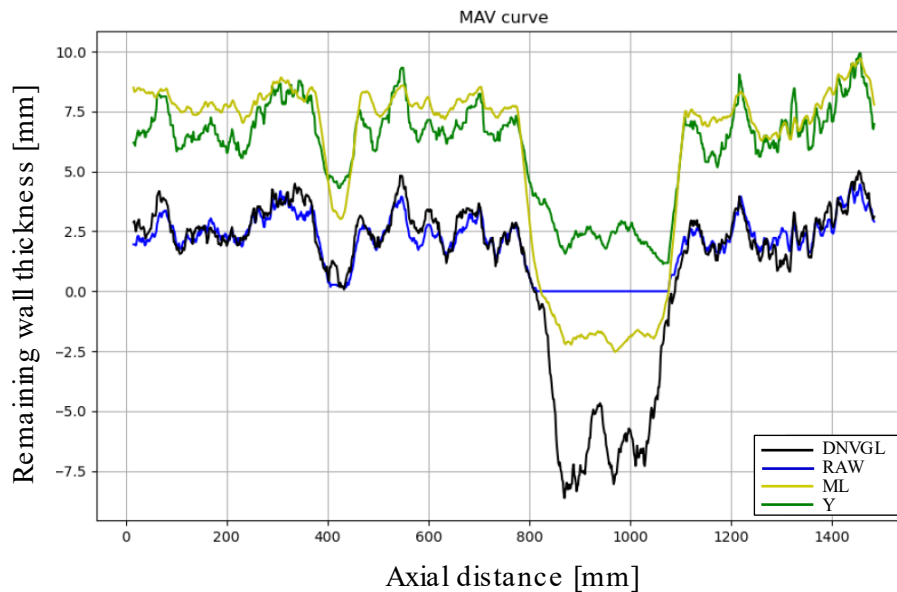
The ML model is trained to filter out the whole noise spectrum, and not only noise outside a certain interval of variation. This means that the ML filter is expected to be less sensitive to minor irregularities than the DNV GL filter. The intention is that these irregularities should be filtered out due to their insignificance to the overall results. The MAV curve generated from RWT predictions from the ML and the DNV GL are plotted in Figure 12. The MAV plot shows that the ML filter has removed noise to a greater extent than the DNV GL filter. The MAV values for the ML filter is close to the Y values and mostly above the Y values. DNV GL has not removed much noise, therefore, estimating RWT values close to the original WT values, which is still on the conservative side. However, the results obtained by the current ML model may be quite optimized and shall be used with cautions.

### 3.3. Simulated pitting corrosion

The wall thickness is set as zero in the pitting area, which is very idealistic and just for test purpose. It is seen from Figure 13 that the ML filter prediction is quite close to the simulated Y-value without losing conservatism in this case. However, the DNV GL filter stays close to the WT data. Negative values are both predicted from the two methods, which is mainly due to the low-quality data odometer between 800 to 1100 mm (the Raw data value for this interval have zero readings as seen in Figure 13). In this case, the ML model provides better predictions, showing that the pitting corrosion patten is recognized.



**Figure 12.** MAV for DNV GL and ML generated from the section with channelling corrosion.



**Figure 13.** MAV plot with DNV GL, ML and Y for the section containing pitting.

#### 4. Conclusions

Following conclusions are made:

- A CNN based machine learning framework has been proposed and established. The simulated data shows that some corrosion patterns have been successfully recognized.
- The performance seen in the case study shows that it is possible to use supervised learning with simulated data to train a model and predict the remaining wall thickness based on real ILI data.
- The confidence in the model's ability to handle different cases could be improved by testing on additional datasets, for example with more fabricated corrosion patterns.

It is expected that the method presented in this paper can be further applied to process real field collected ILI data in the future. It is seen that the machine learning based method may give 'better' results. However, the users must be cautious to use the 'better' results since they may give 'non-conservative' estimation of the remaining wall thickness. Nevertheless, the machine learning-based method presented in this study can be further improved and may be finally used in real projects.

#### Acknowledgement

This work is originated from a master thesis project between NTNU and Aker Solutions AS. The first author would like to thank for the supports from Dr. R. Igland at Aker Solutions AS and Mr. O.E. Lindøe at Equinor (Kårstø) during the project execution.

#### References

- [1] Norwegian Petroleum Directorate, [www.norskpetroleum.no](http://www.norskpetroleum.no), accessed on August, 2021.
- [2] Ma Q, Tian G, Zeng Y, Li R, Song H, Wang Z, Gao B and Zeng K 2021 Pipeline In-line Inspection Method, Instrumentation and Data Management. *Sensors*, **21**(3862). <https://doi.org/10.3390/s21113862>.
- [3] Klann M and Beuker T 2006 Pipeline inspection with the high resolution EMAT ILI-tool: report on field experience. *Proc. of 6<sup>th</sup> Int. Pipeline Conf.* (Calgary, Alberta, Canada. September 25-29, 2006).
- [4] API 2013 In-line Inspection Systems Qualification. API Standard 1163. Second Edition, April 2013.
- [5] DNV GL 2019 Corroded pipelines, DNVGL-RP-F101. Edition September 2019.
- [6] Hennig T, Palmer J, Teves Jan-Eric and Skodzic C 2012 UT ILI river bottom feature assessment of challenging datasets. *Proc. of 7<sup>th</sup> Pipeline Tech. Conf.* (Hannover, Germany, March 28-30, 2012).

- [7] Liu Z, Kvarme S O, Lindøe O E 2018 A comparison study on processing ILI data with different filtering methods. *Proc. of 37<sup>th</sup> Int. Conf. on Offshore Mechanics and Arctic Eng.* OMAE2018-77006. (Madrid, Spain, June 17-22, 2018).
- [8] Barbian A, Beller M, Hugger A, Jaager C and Pfanger A 2010 Thresholds, accuracies and resolutions: quantitative measurement and its advantages for metal loss inspection. *Proc. Of 5th Pipeline Technology Conference* (Hannover, Germany. April 19-20, 2010).
- [9] Van Rossum G. and Drake F L 2009. Python 3 Reference Manual, Scotts Valley, CA: CreateSpace.
- [10] Chollet F et al., 2015 Keras, <https://github.com/fchollet/keras>, 2015.
- [11] Abadi Mart et al., 2016. Tensorflow: A system for large-scale machine learning. *Proc. of 12th USENIX Symposium on Operating Systems Design and Implementation* (OSDI 16). pp. 265–283.
- [12] Thakkar V, Tewary S and Chakraborty C 2018 Batch normalization in convolutional neural networks – a comparative study with cifar-10 data. *Proc. of 5<sup>th</sup> Int. Conf. on Emerging Applications of Information Tech.* (EAIT), 2018, pp. 1-5. DOI: 10.1109/EAIT.2019.8470438.
- [13] Mitchell T M 1997 Machine Learning. 1st ed. USA: McGraw-Hill, Inc., 1997. ISBN: 0070428077.