Candidate Oil Spill Detection in SLAR Data
A Recurrent Neural Network-based Approach

Sergiu-Ovidiu Oprea[1], Pablo Gil[2], Damian Mira[1] and Beatriz Alacid[1]
[1]*University Institute for Computer Research, University of Alicante, San Vicente del Raspeig, Spain*
[2]*Department of Physics, Systems Engineering and Signal Theory, University of Alicante, San Vicente del Raspeig, Spain*
{*sergiu.oprea, pablo.gil, damian.mira, bea.alacid*}*@ua.es*

Keywords: Oil Spill Detection, Maritime Surveillance, SLAR Remote Sensing, RNN, LSTM, Classification.

Abstract: Intentional oil pollution damages marine ecosystems. Therefore, society and governments require maritime surveillance for early oil spill detection. The fast response in the detection process helps to identify the offenders in the vast majority of cases. Nowadays, it is a human operator whom is trained for carrying out oil spill detection. Operators usually use image processing techniques and data analysis from optical, thermal or radar acquired from aerial vehicles or spatial satellites. The current trend is to automate the oil spill detection process so that this can filter candidate oil spill from an aircraft as a decision support system for human operators. In this work, a robust and automated system for candidate oil spill based on Recurrent Neural Network (RNN) is presented. The aim is to provide a faster identification of candidate oil spills from SLAR scanned sequences. So far, the majority of the research works about oil spill detection are focused on the classification between real oil spills and look-alikes, and they use SAR or optical images but not SLAR data. Moreover, the overall decision is usually taken by an operator, mainly due to the wide variety of types of look-alikes which cause false positives in the detection process using traditional NN. This work provides a RRN-based approach for candidate oil spill detection using SLAR data in contrast with the traditional Multilayer Perceptron Neural Network (MLP). The system is tested with time series data acquired from a SLAR sensor mounted on an aircraft. It achieves a success rate in detecting of 97%.

1 INTRODUCTION

Illegal pollution seriously damages marine ecosystems health and induces important scientific political concerns. Oil spill caused by the explosion of Deepwater Horizon oil rig is considered the largest accidental marine oil spill in the history of petroleum industry. Nevertheless, half of the total oil spills in marine ecosystems are caused by intentional discharges (e.g. tank cleaning). It has been estimated that 457,000 tonnes of oil are released in the ocean by shipping every year (GESAMP, 2007)

Oil spill detection by continuous monitoring via satellite or equipped aircraft is a crucial task in order to reduce pollution indices. Synthetic Aperture Radar (SAR) operated on satellites and mounted on aircraft such as Sideward Looking Airborne Radar (SLAR), can be effectively used for this purpose. The interest in this particular research field is limited due to the lack of public SAR and SLAR image datasets. The main step in oil spill detection is performed by trained operators and consists in visual inspection techniques and analysis of extracted features from both images and data. Nevertheless, due to the effectiveness of machine learning-based techniques on remote sensing, semi-automatic or fully automatic approaches are the state-of-the-art in oil spill detection (Topouzelis, 2008). Most of these automatic approaches are related to traditional Multilayer Perceptron (MLP) neural networks, probabilistic approaches and fuzzy classification, using substantial datasets for training and validation.

In image, oil slick detection seems to be trivial for human operators, both semi-automatic and automatic approaches have significant difficulties. Oil slick contrast is a variable feature which depends on oil type, shape, age and as well on weather conditions and ocean tides. Moreover, a wide range of look-alikes such as fish shoals or seaweed accumulations, hinder the detection process (Alacid and Gil, 2016).

During an emergency mission, SLAR position relative to the target varies in time with the aircraft movement. A high resolution image is obtained before processing the successive recorded radar echoes

Oprea, S-O., Gil, P., Mira, D. and Alacid, B.
Candidate Oil Spill Detection in SLAR Data - A Recurrent Neural Network-based Approach.
DOI: 10.5220/0006187103720377
In *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2017)*, pages 372-377
ISBN: 978-989-758-222-6

(Stimson, 1998). In other words and from the machine learning field viewpoint, we are dealing with time series data. For this reason, Recurrent Neural Networks (RNN) can be a feasible solution for this problem (Williams and Zipser, 1989). SLAR-based remote sensing of oil spills in contrast with satellite detection, can cover narrow swaths, identifies the polluter and also determines oil type, amount and if clean-up is necessary. The motivation and main purpose of this work is the development of an automatic candidate oil spill detection system which will be operated under an Aerial Vehicle (AV). A RNN-based approach in SLAR imagery will be presented in contrast to the traditional MLP and other machine learning classification techniques.

Exposed the motivation of this work, the rest of document is structured as follows: in Section 2 a review of related works is performed. Section 3 describes the used dataset and different machine learning-based approaches, analyzed in this work. The methodology followed for the experimentation carried out is shown in Section 4. Finally, Section 5 details the conclusions and draws future work.

2 RELATED WORKS

In spite of the limited literature on oil spill detection using machine learning techniques, there are several relevant research papers. Statistical classifiers based on probabilities are the most known (Solberg et al., 1999) (Fiscella et al., 2000). Fuzzy classification approaches such as (Keramitsoglou et al., 2006) and (Karathanassi et al., 2006) have been also employed successfully. Nevertheless, we will focus on Neural Network-based (NN) methods in order to avoid the feature extraction step considered important in the aforementioned classifiers.

A MLP neural network based-approach with two hidden layers was introduced by (Del Frate et al., 2000). A three-stage pipeline is described: dark spot detection (performed manually as a visual inspection), feature extraction (11 feature vector size describing the dark spots) and classification into oil spill or look-alike using the MLP. This semi-automatic system was trained and tested using 600 low resolution SAR images from the European Remote Sensing (ERS) satellites. Moreover, a pruning procedure was applied to the MLP in order to eliminate ineffective connections. This method, using the leave-one-out approach, misclassified 18% of the oil spills and 10% of the look-alikes.

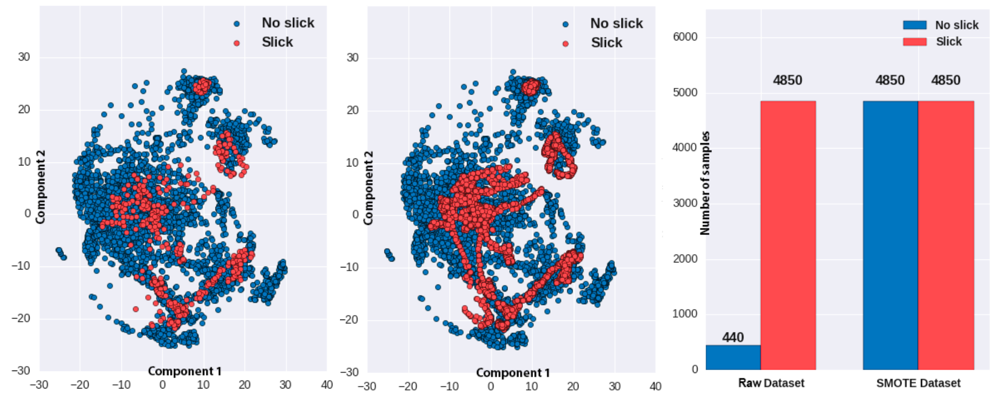
Another MLP neural network approach with one hidden layer (51 neurons), 10 feature input vector size and 2 output nodes was introduced by (Topouzelis et al., 2007). The system was trained and tested using 24 high resolution SAR images containing 69 oil spills and 90 look-alikes. NN topology was configured using a genetic algorithm. The accuracy reported on the test data was: 91% for oil spills and 87% for look-alikes.

A new approach to SAR oil spill detection using two Artificial Neural Network (ANN) in sequence was proposed by (Singha et al., 2013). As a typical SAR-based oil spill detection process (Topouzelis, 2008), a three-stage pipeline was implemented: dark spot detection (first ANN with one hidden layer), feature extraction and classification into oil spill or look-alike (second ANN with two hidden layers). Substantial SAR image dataset from European Maritime Safety Agency (EMSA) was used for training and validation reporting the 91.6% of oil spills correctly classified. A recent comparative study of different classification techniques using RADARSAT-1 SAR imagery (Xu et al., 2014) shows that ANN was the worst classifier among 7 different popular statistical and machine learning classification techniques, such as Support Vector Machine (SVM), tree-based ensemble classifiers (bagging, bundling and boosting), Generalized Additive Model (GAM) and Penalized Linear Discriminant Analysis (PLDA). The tree-based ensemble classifiers obtained more reliable and accurate results in oil spill classification. Using a reduced dataset, PLDA was considered a safer alternative in contrast to more flexible classifiers such as Boosting, ANN or SVM which were prone to cause overfitting. Nevertheless, by applying data standardization and log-transformation regarding to ANN and SVM respectively, performance has been improved.

A comparison in term of classification accuracies between the mentioned classifiers would not be reliable due to the use of different datasets which are not always available, arbitrary number of extracted features dependent on the acquisition sensor, as well as various classifier configurations. For that, we will use raw data for the classification and perform an extensive experimentation with different ANN and RNN configurations over the same dataset (acquired from an aircraft) to achieve a reliable comparison among different techniques based on MLPs and RNNs.

3 METHODOLOGY

The proposed system was implemented using Keras v1.0.8 (Chollet, 2015) running on top of Tensorflow v0.10.0 (Abadi et al., 2015). Keras is a minimalist, highly modular neural network library written in



(a) Raw dataset (b) SMOTE dataset (c) Class distribution

Figure 1: Sample distribution representation using Principal Component Analysis (PCA) (Jolliffe, 2002) over 1158 sized vectors of the original dataset and dataset with SMOTE.

Python and developed for fast prototyping and experimentation. It supports different NN models such as MLPs and RNNs. Moreover, it is easily configurable and runs over both CPU and GPU.

The aim of our system is to classify the inputs in slick and no slick candidates. The output is a binary classification. To accomplish it, we use a single neuron as output layer with a sigmoid activation function and train our system using binary cross entropy as loss function with Adam optimizer (Kingma and Ba, 2014) for MLPs and RmsProp (Tieleman and Hinton, 2012) in the case of RNNs. Input and output sequences are differently processed regarding the RNN models. We use a many to one model where the last sample of a given sequence is classified using the previous computation over the rest of the sequence. In other words, only the last output o_{t+1} of the unrolled network is considered for classification as shown in Figure 3.

In this section we will describe the related problems of our dataset and possible solutions provided by authors, as well as the used machine learning-based techniques and finally the performance appraisal process of our models.

3.1 Dataset

Our system was trained and tested using 12 SLAR records acquired from an aircraft (e.g. Figure 2). Small datasets hinder the application of most machine learning techniques and lead the classifier to overfitting over the training and validation data. Moreover, data noise is considered an significant issue. In order to mitigate these problems, an analysis of our dataset was performed.

Our system inputs are based on the rows of each SLAR record (1158 sized vectors) which represents

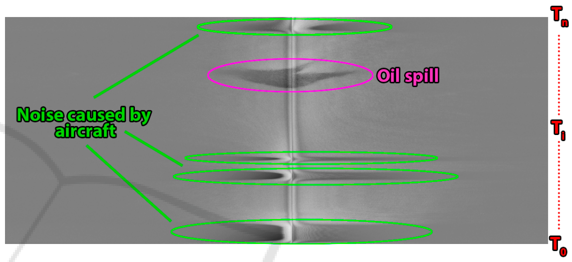


Figure 2: SLAR record representing a video sequence from time t_0 to time t_n where t_i is considered an individual row of data. Oil slick is surrounded by purple and different noise patterns, because of the AV twists, by green.

the SLAR scanning for a time t_i. Dealing with time series data, a row is considered a scanning of a sea part, relied to the aircraft movement. We consider each row as an individual sample. Our system was tested over a balanced dataset of 9700 samples (Figure 1b) where Synthetic Minority Oversampling Technique (SMOTE) was applied over the original dataset of 5290 samples (Figure 1a). SMOTE (Chawla et al., 2002) is an over-sampling approach in which the distance between the feature vector (sample) and its nearest neighbor is computed. The resulted difference is multiplied by a random number ([0,1] range) and added to the feature vector under consideration. In this way, a new different sample is obtained.

Class distribution from raw dataset showed in Figure 1c, represents a 11:1 sample ratio of no slick to slick classes. This indicates that we are dealing with an imbalanced dataset. A dataset is imbalanced when the classification categories are not approximately equally represented. Therefore, an accuracy metric is not appropriate when data are imbalanced and more performance metrics and techniques such as, precision and recall, Receiver Operating Characteristic (ROC) curve and confusion matrix are needed (Chawla, 2005).

In our dataset, different noise types are given. Turns or changes of direction caused by AV (Figure 2) can be successfully removed with a combination of different image processing techniques (Alacid and Gil, 2016). Nevertheless, in this work we train the system with raw data without preprocessing. Using a reduced dataset, noise is considered relevant information for the classifier in order to improve generalization over new input data.

Here, the system is tested from a manually selected dataset with 109 slick and 394 no slick samples.

3.2 Neural Algorithms

MLPs have proven to be very effective in the classification of remote-sensing data reporting outstanding performance results using different configurations. MLP is a feedforward ANN consisting of multiple neuron layers (directed graph structure) with the main purpose of mapping the input data to a set of outputs. It is a modification of linear perceptron in order to classify not linearly separable data.

In MLP and traditional neural network models, all inputs and outputs are independent each other. In contrast with this idea and because of the directed cycle connection between units, RNN are able to create internal states in order to process input sequences. The main idea of RNN is to perform the same task over each element of the sequence depending on previous computation.

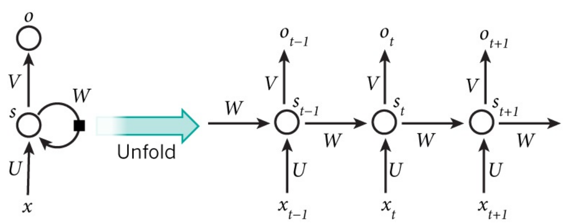


Figure 3: RNN basic structure and unfolding in time according to the number of sequence elements. In this figure, the network is unrolled into a 3-layer NN.

In Figure 3, x_t is the input at time step t. Hidden units are grouped under node s_t and get inputs from other neurons at previous time steps. In this way, RNN can map an input sequence with elements x_t into an output sequence with elements o_t, depending on all the previous x'_t (for $t' \leq t$) (LeCun et al., 2015). In theory, RNNs input sequences can be arbitrarily long, but in practice they are limited to only a few steps (vanishing gradients problem) (Bengio et al., 1994).

The problems of long-term dependencies and vanishing gradients have been solved with Long Short Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997). In contrast with RNNs, unit internal structure has four layers interacting in a special way, instead of heaving a single NN layer. LSTMs help preserve the error which can be backpropagated through time and layers.

There are more sophisticated RNN-based models, such as Bidirectional RNNs (BRNN) (Schuster and Paliwal, 1997). In that model, output at time t also depends on future elements. BRNN structure is just based on two RNNs stacked on top of each other. The main idea is that output is computed regarding hidden states of both RNNs.

A special highlight of our implementations is the use of dropout (Srivastava et al., 2014). Due to the large number of features and the reduced dataset, over-fitting is a real problem to deal. Dropout is a technique for addressing and handling this problem. The key idea is to randomly drop units (along with their connections) from the neural network during training.

3.3 Performance Evaluation

Our experiments were focused into an exhaustive configuration of MLP and RNN solutions reporting the performance results in terms of accuracy, precision and recall using the described dataset in Section 4. Comparing the results we are able to select the most suitable model for our purposes. We have tested the following model configurations with and without dropout on fully connected layers and recurrent connections:

- MLPs with one or two hidden layer(s)
- Vanilla RNN
- LSTM networks
- Bidirectional LSTM networks

Each of the models were tested varying the activation functions on layers (ReLu and sigmoid combinations), neuron number on hidden layer(s), batch size, dropout value and time step length in the case of RNNs. A ranking of best model configurations was performed.

4 RESULTS AND DISCUSSION

After an exhaustive test of multiple different MLP and RNN configurations, a ranking of the best five models of both MLP and RNN networks is represented on

Table 1: Ranking of the five best models of MLPs and RNNs in comparison with C-Support Vector Classification (SVC) with RBF kernel.

Model id	Time steps	Hidden layers	Hidden neurons	Dropout connection	Dropout value	Accuracy (%)	Precision	Recall
BRNN1	3	-	180	recurrent	0.2	97.00	0.9632	1.0
BRNN2	3	-	320	no dropout	-	97.00	0.9701	0.9923
MLP1	-	1	260	input-hidden	0.6	96.82	0.9725	0.9873
MLP2	-	2	240	input-hidden	0.4	96.62	0.9724	0.9847
LSTM1	2	-	140	input-recurrent	0.4	96.22	0.9628	0.9898
MLP3	-	2	100	no dropout	-	96.22	0.9652	0.9873
LSTM2	2	-	280	recurrent	0.2	96.21	0.9722	0.9796
BRNN3	3	-	260	input-recurrent	0.2	96.20	0.9583	0.9949
MLP4	-	2	320	hidden-hidden	0.2	96.02	0.9606	0.9898
MLP5	-	1	260	input-hidden-hidden	0.4	95.63	0.9673	0.9771
SVC	-	-	-	-	-	95.03	0.9424	0.9974

Table 1. In order to a better understanding of the table, a description of each column is required:

- Model id: each network has an identifier representing the NN type
- Time steps (only for RNNs): number of samples of each input sequence. Although, the experiments have been designed with up to 10 time steps, with more than 3 steps, no significant improvement was noticed.
- Hidden layers (only for MLPs): number of hidden layers. MLPs with more than 2 hidden layers have not report a significant improvement and only increased training time.
- Hidden neurons: number of hidden layer(s) neurons. Our system was tested with a number of neurons from 20 to 400, with an increment of 20.
- Dropout connection: indicates where dropout technique is applied (e.g. input-recurrent implies a dropout between inputs and first RNN node and between recurrent connections)
- Dropout value: a 0.4 dropout indicates 40% less connections. Dropout value is selected experimentally, nevertheless, experiments proven that good values are between 0.2 and 0.6 (Srivastava et al., 2014).

Additionally, the results are shown in terms of:

- Accuracy: classification score for correctly predicted samples.
- Precision: measure of result relevancy which relates to a low false positive rate.
- Recall: measure of the number of relevant results returned. Relates to a low false negative rate.

A system with high precision and recall indicates that the classifier is returning many results (high recall) with all results labelled correctly (high precision). All tested configurations pointed to good classification results in terms of accuracy, precision and recall. Nevertheless, a better general performance has been achieved with RNN configurations, concretely BRNN1 and BRNN2. The small performance difference between RNNs and MLPs indicates that both NN models fitted very well our SLAR dataset. In order to avoid overfitting, early stopping technique regarding validation loss has been applied. In the implemented experiments, our system stops the training when the validation loss value stops its decreasing. Dataset and model complexity should be directly proportional. Otherwise, the machine learning model will overfit over both validation and test data. In order to ensure that a simpler model underperformed that neural networks, a SVC model was implemented. Classification results were above 94% of accuracy.

LSTM and BRNN performed better than vanilla RNNs. For that reason, vanilla RNNs are not present in the ranking. The reduced time steps number regarding RNNs should be due to noise presence in our input data. Experimental results show that our systems perform better with as much 3 time steps sequences.

5 CONCLUSIONS

In this paper different RNN-based machine learning techniques have been implemented and tested for candidate oil spill detection using SLAR data acquired from an AV. BRNN with two different configurations have reported the best classification results from our test dataset (97% accuracy). A general better performance was achieved using RNNs instead of MLPs whose use is widespread in most of the state-of-the-art works regarding this issue. In order to overcome the imbalanced dataset problem, SMOTE technique was successfully applied. The use of RNN in this paper was mainly motivated by the naturalness of SLAR data.

This work is considered an initial approach for a robust candidate oil spill detection system using SLAR data with the main purpose of achieving a faster identification of the polluter ship. As future work, advanced LSTM networks variations such as Gated Recurrent Unit (GRU) (Cho et al., 2014) will be tested. At the same time, more data will be provided to keep training our system for achieving more robustness.

ACKNOWLEDGEMENTS

The authors would like to thank INAER Helicopters SAU, which is part of Babcock International Group plc, for the provision of data. This work was supported by the Spanish Ministry of Economy and Competitiveness through the research project ONTIME (RTC-2014-1863-8).

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2015). Tensorflow: Large-scale machine learning on heterogeneous systems, 2015. *Software available from tensorflow. org*, 1.

Alacid, B. and Gil, P. (2016). An approach for slar images denoising based on removing regions with low visual quality for oil spill detection. *SPIE Remote Sensing, Image and Signal Processing for Remote Sensing XXII*, 10004(19):1–10.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Chawla, N. V. (2005). Data mining for imbalanced datasets: An overview. In *Data mining and knowledge discovery handbook*, pages 853–867. Springer.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority oversampling technique. *Journal of artificial intelligence research*, 16:321–357.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chollet, F. (2015). Keras. *GitHub repository: https://github. com/fchollet/keras*.

Del Frate, F., Petrocchi, A., Lichtenegger, J., and Calabresi, G. (2000). Neural networks for oil spill detection using ers-sar data. *IEEE Transactions on geoscience and remote sensing*, 38(5):2282–2287.

Fiscella, B., Giancaspro, A., Nirchio, F., Pavese, P., and Trivero, P. (2000). Oil spill detection using marine sar images. *International Journal of Remote Sensing*, 21(18):3561–3566.

GESAMP, J. G. o. E. o. t. S. A. o. M. E. P. (2007). Estimates of oil entering the marine environment from sea-based activities.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Jolliffe, I. (2002). *Principal component analysis*. Wiley Online Library.

Karathanassi, V., Topouzelis, K., Pavlakis, P., and Rokos, D. (2006). An objectoriented methodology to detect oil spills. *International Journal of Remote Sensing*, 27(23):5235–5251.

Keramitsoglou, I., Cartalis, C., and Kiranoudis, C. T. (2006). Automatic identification of oil spills on satellite images. *Environmental Modelling & Software*, 21(5):640 – 652.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Singha, S., Bellerby, T. J., and Trieschmann, O. (2013). Satellite oil spill detection using artificial neural networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 6(6):2355–2363.

Solberg, A. S., Storvik, G., Solberg, R., and Volden, E. (1999). Automatic detection of oil spills in ers sar images. *IEEE Transactions on Geoscience and Remote Sensing*, 37(4):1916–1924.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Stimson, G. W. (1998). *Introduction to airborne radar*. SciTech Pub.

Tieleman, T. and Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2).

Topouzelis, K., Karathanassi, V., Pavlakis, P., and Rokos, D. (2007). Detection and discrimination between oil spills and look-alike phenomena through neural networks. {*ISPRS*} *Journal of Photogrammetry and Remote Sensing*, 62(4):264 – 270.

Topouzelis, K. N. (2008). Oil spill detection by sar images: dark formation detection, feature extraction and classification algorithms. *Sensors*, 8(10):6642–6659.

Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Xu, L., Li, J., and Brenning, A. (2014). A comparative study of different classification techniques for marine oil spill identification using radarsat-1 imagery. *Remote Sensing of Environment*, 141:14–23.