SMARTSAT
COOPERATIVE RESEARCH CENTRE

TECHNICAL REPORT 4

Development of Machine Learning and Deep Learning Models for Classification of Radar Signals

TECHNICAL REPORT 4

Development of Machine Learning and Deep Learning Models for Classification of Radar Signals

NOVEMBER 2023

This report should be cited as:
SmartSat 2021, Development of deep learning models for simulated radar datasets, SmartSat Technical Report 4, SmartSat, Adelaide, Australia.

Disclaimer:
This publication is provided for the purpose of disseminating information relating to scientific and technical matters. Participating organisations of SmartSat do not accept liability for any loss and/or damage, including financial loss, resulting from the reliance upon any information, advice or recommendations contained in this publication. The contents of this publication should not necessarily be taken to represent the views of the participating organisations.

Acknowledgement:
SmartSat acknowledges the contribution made by DEWC Systems and Deakin University towards the writing and compilation of this technical report.

Executive Summary

Project P2-20 is a collaboration between *DEWC Systems* and the *Institute for Intelligent Systems Research and Innovation (IISRII), Deakin University*. The main project objective is to design and develop intelligent data-based models for classification of radar signals with potential distributed radio frequency (RF) processing capabilities for deployment across space-based platforms.

The project leverages advancements in artificial intelligence (AI) to develop cutting edge intelligent data analytic solutions for radar monitoring capabilities. This initial collaboration is part of a multi-phase AI-driven RF monitoring system roadmap of DEWC systems to advance sovereign RF technologies and capabilities in Australia, along with the support of SmartSat CRC. The collaboration team brings together researchers from Deakin University and domain experts from DEWC Systems to analyse and address radar signal analytics, leading to detection and classification of conventional and low-probability-of-intercept (LPI) radar signals.

The eight-month duration of the project has resulted in a proof-of-concept deep learning-based framework capable of detecting and classifying synthetic RF signals tested in a lab environment with simulated noise. A variety of data sets with different signal-to-noise ratios (SNRs) from different target radar signals have been generated for evaluation. The obtained accuracy scores across different scenarios are promising, yielding close to perfect classification performance and demonstrating efficacy of the developed DL framework for RF signal detection and classification.

Table of Contents

Table of Contents

Table of Figures

Table of Tables

1. Introduction

Two machine learning (ML) and deep learning (DL)-based pipelines were fully designed and implemented as part of this project. These two pipelines differ in how they transform and process time series data (radar echoed signals), perform feature extraction and develop classification models. Details of these two pipelines along with the obtained results are discussed.

2. ML Pipeline

2.1 Preliminary

To build the ML pipeline, a binary classification is formulated for analyzing data generated based on characteristics with respect to radar systems. Data samples in conducted experiments in this section are radar echoes from a cylinder and a cone. These samples are generated using the Radar Toolbox of Matlab.

As an example, the signals in Figure 1 indicate 100 echo returns from a cylinder over time. The assumption data generation is that the cylinder undergoes a motion that causes tiny vibrations around its boresight. This results in the aspect angle changes from one sample to the next.

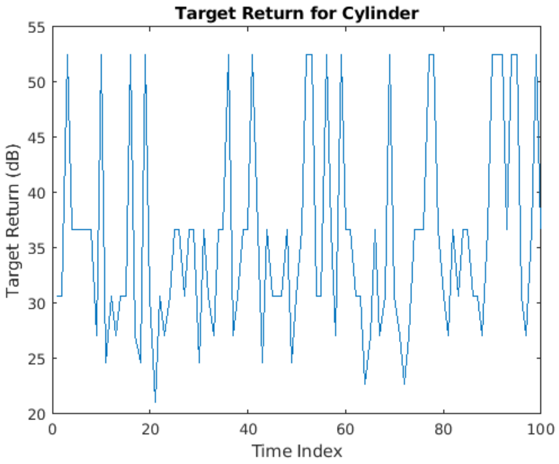


FIGURE 1: THE TARGET RETURN FOR CYLINDER OBJECT.

2.2 Machine Learning (ML) Pipeline

The ML pipeline mainly covers data transformation, feature extraction from time series, and classifier development.

The following subtasks are completed as part of this activity:

- Data generation using Radar Toolbox in Matlab
- Conversion of the data to the CSV format for processing using software tools and packages in Python
- Data manipulation in Python
- Feature extraction in Python using the principal component analysis (PCA)
- Exploratory data analysis based on the extracted features

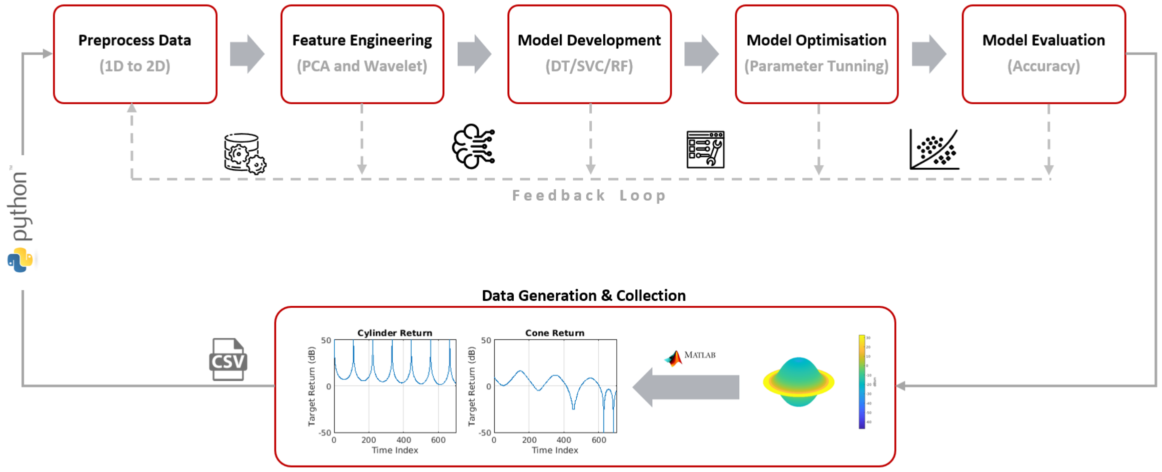


FIGURE 2: THE PIPELINE FOR SIGNAL PROCESSING, FEATURE EXTRACTION, AND MODEL DEVELOPMENT AND EVALUATION USING TRADITIONAL MACHINE LEARNING TECHNIQUES.

- Development of the classification models, which include random forest and decision trees
- Comprehensive data pre-processing, model evaluation and examination

These steps are visually presented in **Figure 2**. The synthetic data generation part has been completed in Matlab. The generated CSV files are imported into a Python software and processed using ML and data processing packages.

2.3 ML Results and Discussion

2.3.1 Data Transformation

The results of data pre-processing, transformation, and feature extraction are shown in Figure 3. The first five principal components contain more than 70% of information variation in the data set, indicating the importance of these features for ML model development.

Figure 4 depicts transformed time series (radar echoed signals) in the 2D space of principal components. Each point corresponds to an echoed signal from a cone or cylinder. There is a very good separation between features extracted using PCA. This separation is obvious for principal components 1 and 2 as well as 1 and 3, i.e. no overlap between density plots of two classes (data from cone and cylinder).

This exploratory data analysis reveals that the data samples from the target classes are well-separated, and existing ML models, such as support vector machines, decision trees, etc., could achieve promising results.

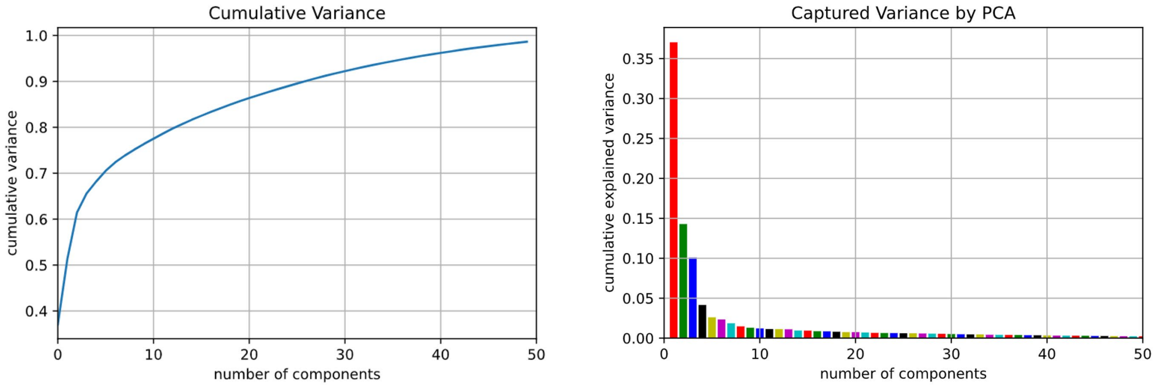


Figure 3: The result of data transformation and processing using principal component analysis. The plot on the left side shows the cumulative variance captured by principal components. The plot on the right side demonstrates the variance captured by individual principal components. These two plots indicate that the first few components effectively and efficiently capture most of the data variance and

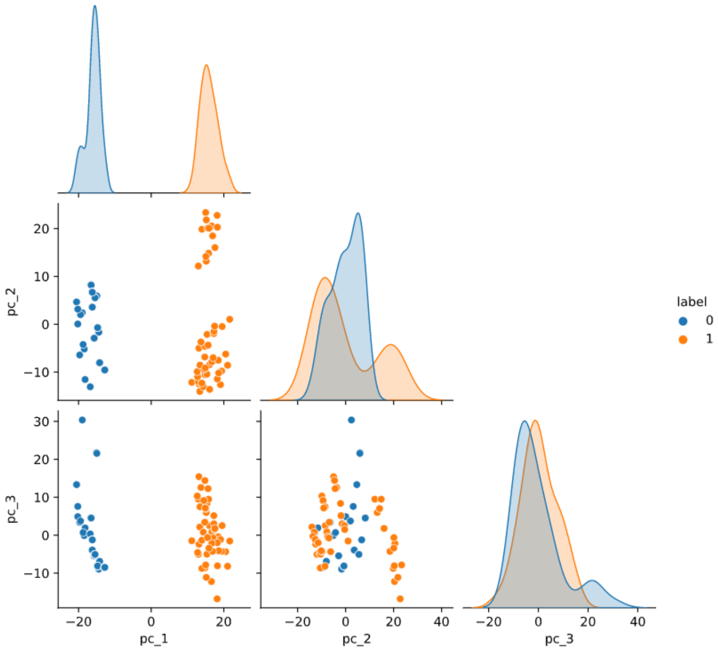


Figure 4: The scatterplot of top PCA features for two classes. A very good separation between these classes is obtained just using features 1 and 2 or 1 and 3.

2.3.2 Classification Results

Based on insights obtained from exploratory data analysis in the feature space, support vector classifiers (SVCs) are identified for ML model development for tackling this binary classification problem. The process of model development and evaluation is as follows:

- Repeat the following steps 100 times:
    - Randomly split the data samples into training (80%) and test (20%) sets.

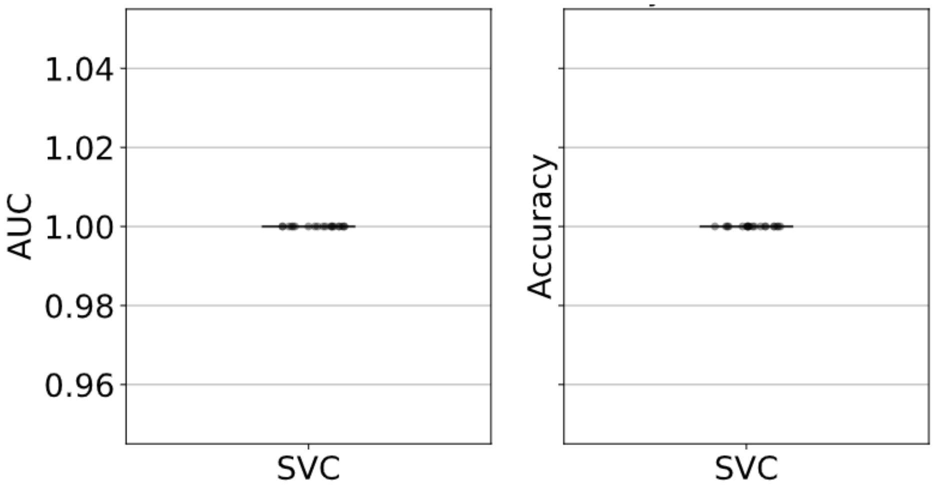


FIGURE 5: AUC AND ACCURACY PERFORMANCE METRICS IN 100 RUNS OF SVC MODEL.

  - Develop an SVC model using the training set
  - Check and record the model performance for the test set (accuracy and area under the Receiver Operating Characteristic (ROC)curve (AUC))
- Generate the density plots for the performance metrics and calculate their statistics.

Figure 5 depicts the boxplots of AUC and accuracy pertaining to the test samples from 100 runs (training and evaluating the model 100 times). SVC models achieve 100% accuracy and AUC of 1 in all experiments. This observation is within expectation considering the scatterplots of time series in the feature space (as shown in Figure 4). The zero variance in the AUC and accuracy scores from 100 runs indicate the effectiveness of the proposed ML pipeline for accurate classification radar signals for this specific case.

The AUC plot for one of the trained SVC models is shown in Figure 6. This plot indicates that the ML model is capable of perfectly separating cone and cylinder echoed signals based on a few key features identified through PCA.

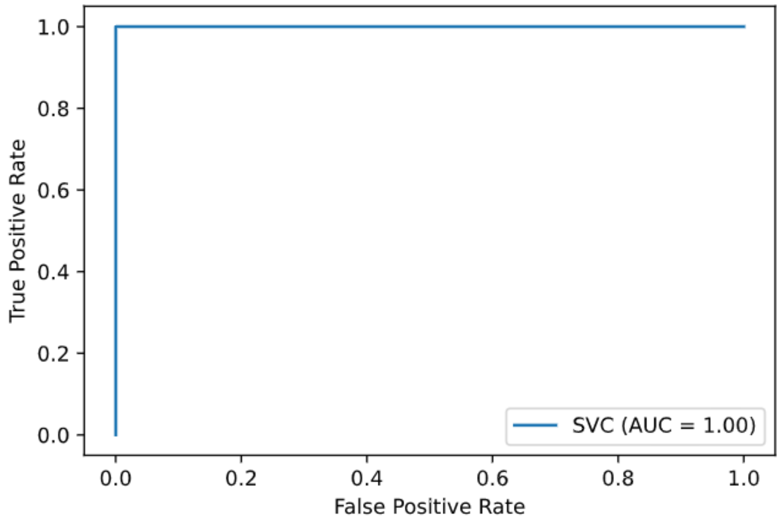


FIGURE 6: THE AUC PLOT OF SUPPORT VECTOR CLASSIFIER IN ONE OF THE TRAINING SESSIONS.

3. Deep Learning Pipeline

3.1 Motivation to Develop a DL Pipeline

Research in Artificial Intelligence (AI) was revolutionised in 2012 by the introduction of deep learning (DL) algorithms [1]. DL realises large artificial neural networks, which are inspired by biological neurons of humans brains. Compared with conventional ML techniques, DL models can achieve best-in class performance, scale effectively with data, automatically extract useful information (i.e., features) for decision making, and are fully transferable [2]. New advances in DL have ignited an explosion of AI applications for solving challenging problems across a range of previously closed applications, as diverse as autonomous cars [3], cancer diagnosis [4] [5], and drug discovery [6].

The proposed ML pipeline described in the previous section achieves promising results under specific conditions:

- A low level of communication noise;
- Data abundance; and
- Binary signal classification.

These assumptions could be easily violated in real-world due to multiple reasons:

- Usually, communication noise is high resulting in corrupted signals. The transformation of and feature extraction from these signals lead to poor data representation.
- It could be the case that the number of collected samples is very limited, e.g. due to technical problems including hardware constraints (e.g. memory shortage).
- The separation of multiple classes becomes practically impossible in the space of features obtained using linear transformations.
- While effective, the procedure of feature engineering requires extensive effort and domain knowledge to yield acceptable classification results.

Considering these issues and shortcomings, we propose and implement a DL pipeline for processing time series signals from radar systems. The key benefits of DL models leveraged in this study are:

- State-of-the-art performance in 1D and 2D classification problems;
- No need for manual feature engineering;
- Possibility of using pre-trained models;
- Possibility of using optimised and proven architectures.

3.2 Data Generation

The data set used for DL experiments was generated based on the information on the following website:

https://au.mathworks.com/help/deeplearning/ug/radar-waveform-classification-using-deep-learning.html

The simulated radar system generates data for the following types of signals:

- Rectangular
- Linear frequency modulation (LFM)
- Barker Code
- Gaussian frequency shift keying (GFSK)
- Continuous phase frequency shift keying (CPFSK)
- Broadcast frequency modulation (B-FM)
- Double sideband amplitude modulation (DSB-AM)
- Single sideband amplitude modulation (SSB-AM)

Figure 7 shows three randomly selected LFM radar waveforms. These plots clearly depict the differences/variances in the generated data samples.

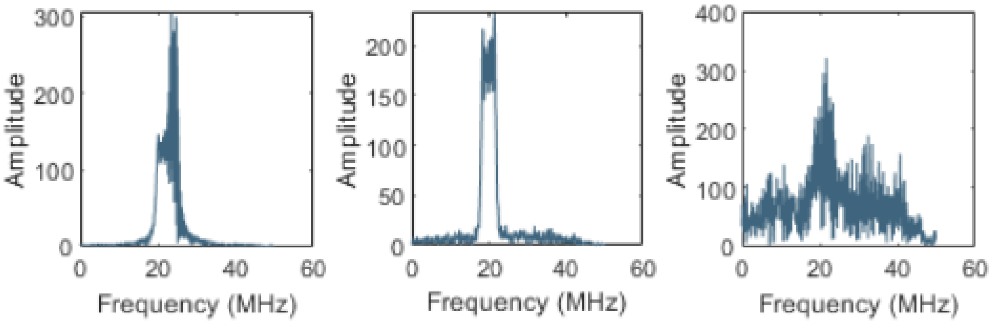


FIGURE 7: RADAR WAVEFORMS GENERATED DURING SIMULATIONS.

We also change the signal to noise (SNR) ratio during the data generation process. This is to simulate scenarios where there is no communication interference (high SNR) and where the communication is corrupted by noise (low SNR). To comprehensively examine the noise impact, we generate multiple data sets for different SNR values, i.e., from -10 to 20 with an increment of 5. These seven data sets are used to train DL models where their performance is evaluated using unseen data samples in the test set.

3.3 1D to 2D Data Encoder

DL models have been mainly developed for 2D data structures such as images. As such, we convert 1D signals into 2D images by applying the Wigner-Ville distribution transformation. This 2D encoder provides an effective representation of the time series data in the time-frequency domain. This representation makes pattern recognition and classification easier for the DL model. The Wigner-Ville distribution generates a time-frequency representation of the original time series data. This transformation is suitable for time varying signals. The high resolution and locality in both time and frequency provide ample information/patterns for the identification of similar modulation types. The obtained 2D views are saved as pictures, which are then used for model development.

Figure 8 shows nine pictures corresponding to nine randomly selected waveforms after their transformation using the Wigner-Ville distribution encoder.

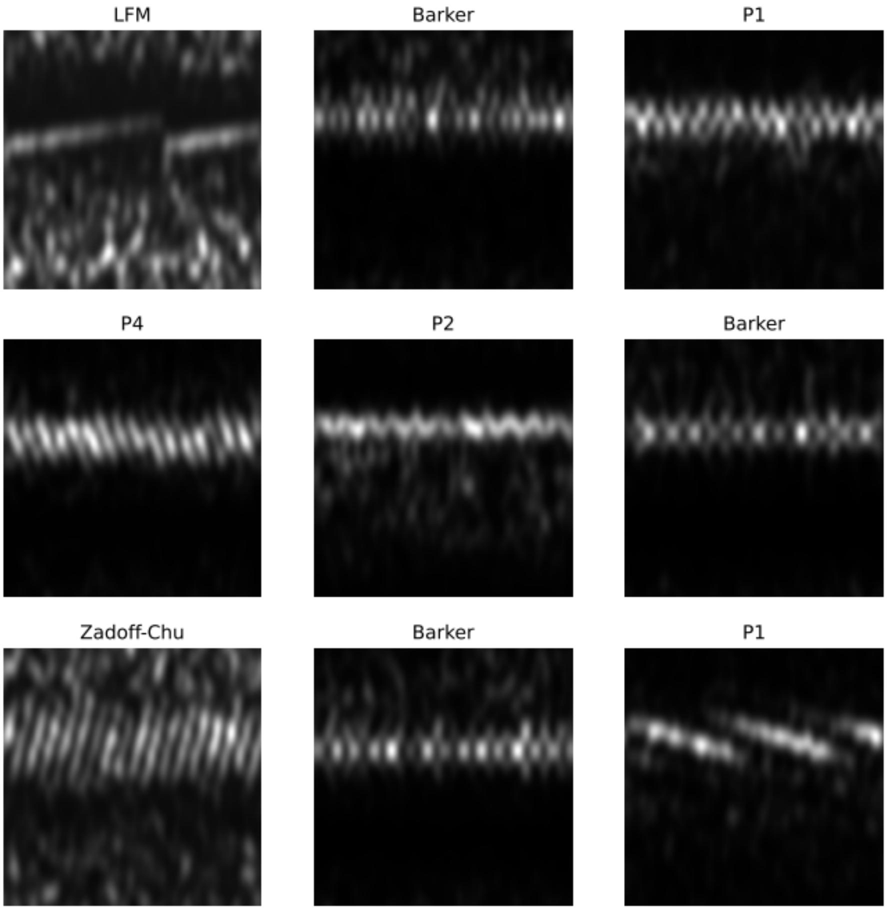


FIGURE 8: NINE RANDOMLY SELECTED WAVEFORM AFTER THEIR 2D TRANSFORMATION.

3.4 DL Model Architecture

The DL pipeline for processing radar waveforms is shown in Figure 9. The 2D images are fed to a deep convolutional neural network (CNN). The model consists of one rescaling layer, three convolutional layers, and one dense layer. The model treats the radar signals as 2D images (non-RGB) and outputs one of the target classes, as per the data generation process.

The model uses ‘Adam’ optimiser for adjusting its parameters through minimisation of the categorical cross-entropy as the loss function. Max pooling is applied after each convolutional layer to calculate the maximum value for patches of a feature map and create a downsampled feature map. The model is trained for only 15 epochs as we found the performance hits the maximum limit with this number of epochs. The dropout mechanism is also applied to the layers of the network to guard against overfitting during the training process.

Figure 10 demonstrates the number of parameters per layer and the total number of DL parameters. Similar to other CNN models used for computer vision tasks, the majority of the model parameters belong to the network head (fully connected part) where 50,176 flattened features are mapped to 128 features.

The model is developed using TensorFlow so we can use distributed training and warm start-up, which allows us to start training using previously optimized model parameters. This allows the use of online learning in a distributed fashion as we are now able to periodically update the model as new data samples are collected.

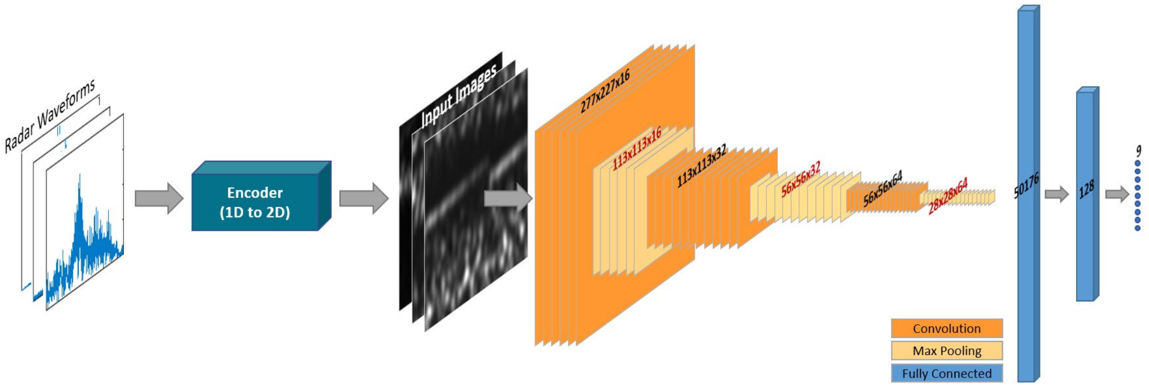
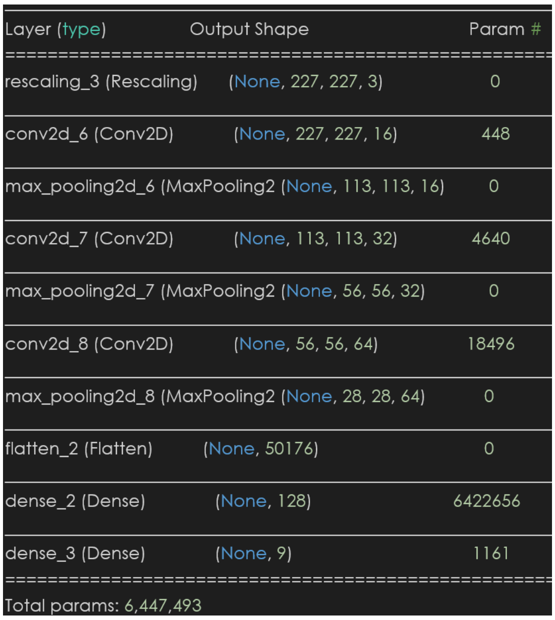


FIGURE 9: DL PIPELINE FOR THE CLASSIFICATION OF RADAR WAVEFORMS.



```
Layer (type)                Output Shape              Param #
=================================================================
rescaling_3 (Rescaling)     (None, 227, 227, 3)       0

conv2d_6 (Conv2D)           (None, 227, 227, 16)      448

max_pooling2d_6 (MaxPooling2 (None, 113, 113, 16)     0

conv2d_7 (Conv2D)           (None, 113, 113, 32)      4640

max_pooling2d_7 (MaxPooling2 (None, 56, 56, 32)       0

conv2d_8 (Conv2D)           (None, 56, 56, 64)        18496

max_pooling2d_8 (MaxPooling2 (None, 28, 28, 64)       0

flatten_2 (Flatten)         (None, 50176)             0

dense_2 (Dense)             (None, 128)               6422656

dense_3 (Dense)             (None, 9)                 1161
=================================================================
Total params: 6,447,493
```

FIGURE 10: THE DEEP CNN ARCHITECTURE AND THE NUMBER OF PARAMETERS PER LAYER.

3.5 DL Model Convergence and Performance

The convergence plots of the DL model are shown in Figure 11. These plots are generated for 10 epochs in one of the training sessions corresponding to an SNR value of -10 (strong noise). The validation set is utilised for detecting when overfitting starts to happen, allowing the use of weights corresponding to that epoch for the final model.

More information about the convergence of DL models for different SNR values is provided in Appendix A.

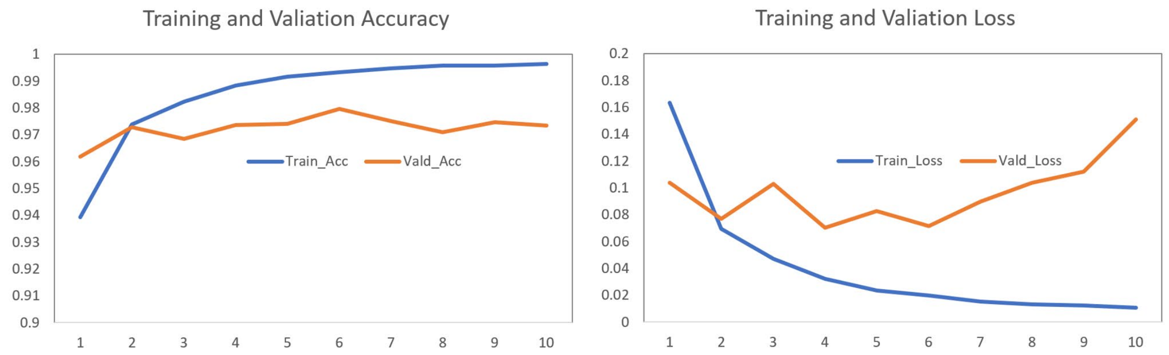


FIGURE 11: THE ACCURACY AND LOSS PLOTS FOR THE TRAINING AND VALIDATION SET DURING THE MODEL DEVELOPMENT. THE RESULTS ARE SHOWN ONLY FOR 10 EPOCHS.

The confusion matrix for the test samples of a data set with an SNR value of -10 is shown in Figure 12. Only a few samples from specific classes (mainly P1 and P2) have been incorrectly classified. This is due to the high similarity of signals for these specific classes.

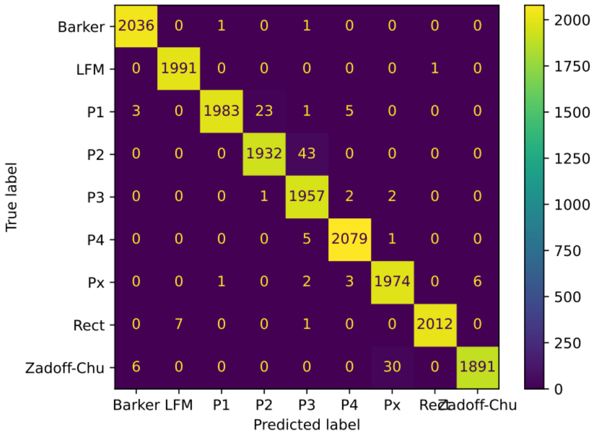


FIGURE 12: THE CONFUSION MATRIX FOR NINE CLASSES OBTAINED FOR THE TEST SET.

3.6 Accuracy vs SNR Plot

The most important chart in this technical report is the accuracy vs SNR values. This is to investigate how communication noise and other interferences impact classification accuracy of DL models pertaining to radar waveforms. A new CNN model is trained using data from each SNR value and its performance is examined for the test set. A total of seven CNN models are constructed in this analysis.

Figure 13 displays obtained accuracy rates for different SNR values. It is observed that:

- The network test accuracy and SNR values are positively correlated. The smaller the SNR value, the smaller the accuracy rate of the test set.
- For SNR values greater than zero, the DL models achieve a near perfect (100%) accuracy score.
- The minimum accuracy score of the test set is 97.97% for the SNR value of -10. This is a very severe case of high communication noise. Despite that, the CNN model achieves promising classification results.

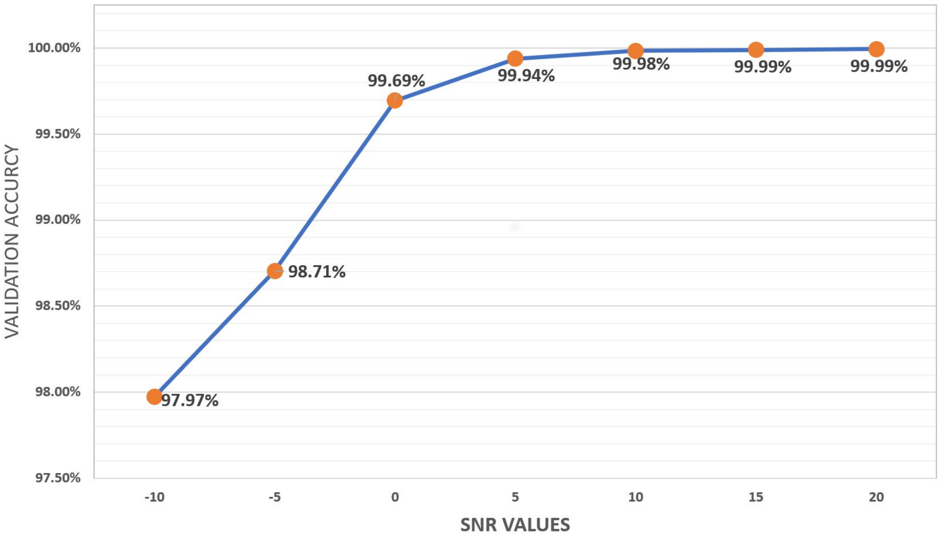


FIGURE 13: THE ACCURACY OF THE TEST SET FOR DIFFERENT SNR VALUES. A NEW DL MODEL IS TRAINED AND TESTED FOR EACH SNR DATASET.

4. Conclusion

The obtained results in this proof-of-concept project indicate that the DL methodology can be effectively and efficiently applied to classification of radar waveforms. Comprehensive simulations using synthetic data sets demonstrate the promising competency of deep CNN models to automatically extract useful features from data and utilise them to correctly categorise radar signals into different groups. The proposed DL solution is also robust against noise and uncertainties that negatively impact the quality of communication signals.

It is important to note that Phase 1 of this project has been focused on developing a DL pipeline for binary classification of synthetic radar signals. Phase 2 of this project has taken a step further to handle multi-class data sets with varying SNR values. Promising results with near perfect classification accuracy have been achieved.

5. Further Research

P2-20 Phase 1 has been a very successful project both in terms of the research results and the level of collaboration achieved by DEWC Systems and the IISRII at Deakin University. This collaboration and the project outcomes have formed a strong basis for further research. In particular, the existing pipeline can be enhanced to undertake the following tasks:

- Process actual radar signals in real-time.
- Develop parsimonious models (minimum computational burden and being able to be trained on small battery-powered GPUs).
- Deploy and test the developed DL solution on a battery-powered GPU system.
- Improve the robustness of models considering high level of communication noise/uncertainty.

6. References

[1] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature,* vol. 521, pp. 436, 05/27/online, 2015.

[2] Y. Bengio, I. Goodfellow, and A. Courville, *Deep Learning*: MIT Press, In preparation, 2015.

[3] M. M. Waldrop, “Autonomous vehicles: No drivers required,” *Nature,* vol. 518, no. 7537, pp. 20-23, 2015.

[4] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, “Dermatologist-level classification of skin cancer with deep neural networks,” *Nature,* vol. 542, no. 7639, pp. 115-118, 2017.

[5] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sánchez, “A survey on deep learning in medical image analysis,” *Medical Image Analysis,* vol. 42, pp. 60-88, 2017.

[6] H. Chen, O. Engkvist, Y. Wang, M. Olivecrona, and T. Blaschke, “The rise of deep learning in drug discovery,” *Drug Discovery Today,* vol. 23, no. 6, pp. 1241-1250, 2018/06/01/, 2018.

Appendix A

Accuracy and Loss for SNR Values

TABLE 1: PERFORMANCE METRICS FOR THE SNR VALUE OF -10

Epoch	Loss	Accuracy	Validation Loss	Validation Accuracy
1	0.163468	0.939194	0.103636	0.961944
2	0.069572	0.97375	0.076991	0.972833
3	0.047218	0.982306	0.103089	0.968389
4	0.032026	0.988278	0.070413	0.973667
5	0.02338	0.991611	0.082567	0.974
6	0.019583	0.993333	0.071448	0.979722
7	0.015223	0.994778	0.089808	0.975
8	0.013349	0.995722	0.103795	0.971056
9	0.012364	0.995861	0.111919	0.974722
10	0.010872	0.996403	0.150795	0.9735

TABLE 2: PERFORMANCE METRICS FOR THE SNR VALUE OF -5

Epoch	Loss	Accuracy	Validation Loss	Validation Accuracy
1	0.13051	0.95383	0.06894	0.97428
2	0.05230	0.98132	0.04120	0.98472
3	0.03463	0.98815	0.08326	0.96911
4	0.02343	0.99179	0.04498	0.98472
5	0.01783	0.99374	0.04154	0.98706
6	0.01208	0.99618	0.04952	0.98589
7	0.00868	0.99714	0.06209	0.98378
8	0.00959	0.99694	0.07679	0.98472
9	0.00948	0.99721	0.07509	0.98378
10	0.00773	0.99751	0.07240	0.98489

TABLE 3: PERFORMANCE METRICS FOR THE SNR VALUE OF 0

Epoch	Loss	Accuracy	Validation Loss	Validation Accuracy
1	0.11788	0.95882	0.06725	0.97422
2	0.03176	0.98914	0.03131	0.98917
3	0.01774	0.99399	0.02546	0.99100
4	0.01346	0.99554	0.02110	0.99328
5	0.01060	0.99660	0.02053	0.99400
6	0.00794	0.99751	0.02672	0.99311
7	0.00566	0.99824	0.03453	0.99094
8	0.00500	0.99844	0.01255	0.99661
9	0.00464	0.99875	0.01307	0.99694
10	0.00629	0.99825	0.03305	0.99194

TABLE 4: PERFORMANCE METRICS FOR THE SNR VALUE OF 5

Epoch	Loss	Accuracy	Validation Loss	Validation Accuracy
1	0.08332	0.97072	0.01177	0.99611
2	0.01768	0.99435	0.01590	0.99489
3	0.01009	0.99697	0.00728	0.99772
4	0.00686	0.99779	0.00454	0.99861
5	0.00483	0.99839	0.00487	0.99906
6	0.00675	0.99797	0.01053	0.99794
7	0.00520	0.99869	0.00655	0.99867
8	0.00010	0.99997	0.00532	0.99928
9	0.00000	1.00000	0.00561	0.99939
10	0.00469	0.99910	0.01195	0.99728

Table 5: Performance metrics for the SNR value of 10

Epoch	Loss	Accuracy	Validation Loss	Validation Accuracy
1	0.075362667	0.974291682	0.011515077	0.99627775
2	0.010332919	0.996652782	0.022697451	0.995666683
3	0.009682666	0.99693054	0.021585153	0.994555533
4	0.004839437	0.99869442	0.008029566	0.998388886
5	0.004596752	0.998680532	0.004434864	0.999111116
6	0.004867876	0.998597205	0.011049566	0.997555554
7	0.000568573	0.999833345	0.001536458	0.999833345
8	0.004430521	0.999000013	0.003521878	0.999722242
9	0.002446122	0.99934721	0.004129854	0.999611139
10	9.98E-06	1	0.004036414	0.999666691

Table 6: Performance metrics for the SNR value of 15

Epoch	Loss	Accuracy	Validation Loss	Validation Accuracy
1	0.07072749	0.975652754	0.007146718	0.997222245
2	0.010374228	0.996708333	0.001994345	0.999277771
3	0.006389605	0.998013914	0.00319334	0.999555528
4	0.000161575	0.999944448	0.001846861	0.999888897
5	0.006560472	0.998472214	0.000724981	0.999833345
6	0.003543315	0.999097228	0.001153432	0.999833345
7	0.004396447	0.998930573	0.003054217	0.999388874
8	0.005304446	0.99869442	0.006278185	0.999111116
9	0.001965454	0.999527752	0.002876327	0.999777794
10	0.004079413	0.999222219	0.002653279	0.999666691

Table 7: Performance metrics for the SNR value of 20

Epoch	Loss	Accuracy	Validation Loss	Validation Accuracy
1	0.069147691	0.97613889	0.015477251	0.996222198
2	0.014174704	0.995930552	0.001411045	0.999555528
3	0.006691303	0.998027802	0.001180926	0.999555528
4	0.002674808	0.999277771	0.000290291	0.999888897
5	0.005282861	0.99862498	0.005270964	0.998666644
6	0.001729035	0.99954164	0.000316811	0.999944448
7	1.05E-05	1	0.000202576	0.999944448
8	1.29E-06	1	0.000206467	0.999944448
9	3.38E-07	1	0.00018567	0.999944448
10	2.04E-07	1	0.00020375	0.999944448

USER INFORMED INDUSTRY DRIVEN RESEARCH POWERED

SMARTSAT
COOPERATIVE RESEARCH CENTRE

Australia's Premier Space Research Centre

Australian Government
Department of Industry, Science and Resources

AusIndustry
Cooperative Research Centres Program

SmartSat CRC Head Office:
Lot Fourteen, Level 2, McEwin Building
North Terrace, Adelaide, SA

info@smartsatcrc.com
smartsatcrc.com