APPENDIX E

TESTING AND RESULTS

TABLE OF CONTENTS

INTRODUCTION

This document outlines setup, testing and results in the three generated datasets.

SETUP

TRADITIONAL METHODS

To implement the traditional approach to deep learning, MATLAB scripts were developed. These scripts can be found in Appendix F.

DEEP LEARNING

To implement the deep learning approaches, much more computational power was need in comparison to traditional methods in order to train the models. Models were trained on the Cypress Semiconductor GPU cluster. The specs of the cluster node used are,

- Nvidia GeForce RTX 2080
- 512GB RAM
- 1 TB SSD

Python was used for the development of the models. In particular, keras with a tensorflow backend. Many different machine learning libraries were also used, including numpy, pandas etc. All libraries used, and code produced can be seen in Appendix E.

DATA

As outlined in Appendix D, the data is generated using m-sequences. The following abbreviations used in the diagrams throughout this document illustrate the following,

- Ref Ref
  - Reference polynomial and reference initial state, label y = 1
- Ref Diff:
  - Reference polynomial and random initial state, label y = 0
- Diff Diff
  - Random polynomial and Random initial state, Label y = 0

TRADITIONAL PACKET DETECTION METHODS

BINARY SYMMETRIC CHANNEL (BSC)

Fig.1 - Fig. 7 illustrate the probability of detection under a range of thresholds based on the given error probability. Fig.1 shows an error probability of 0.2, which is equivalent and close to what is used in industry and therefore used as the error probability to build the training dataset for the deep learning models below. It shows that for ref, ref values at a threshold value of 0 to 0.25, there is 100% probability of detection. With a threshold of 0.95 threshold yields a 0% probability of detection. However, for ref, diff and diff, diff sequences at a threshold value of 0 there is 0.45% and 0.5% chance of detection. WIth a threshold value of 0.4 having 0% detection. This illustration is correct as with random polynomials and random initial states the detection level will decrease quicker.

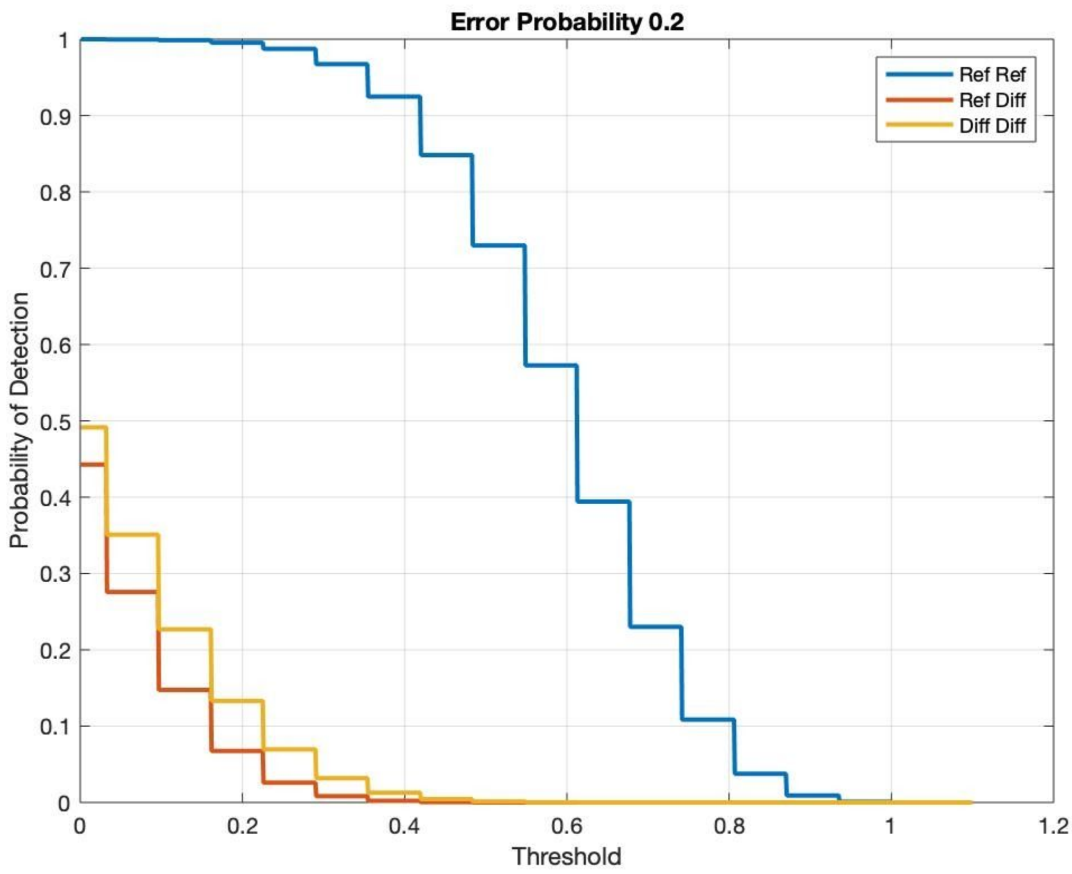


Fig. 1. Error Probability 0.2

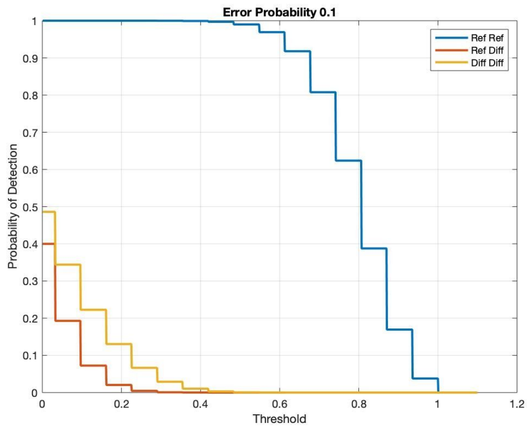


Fig. 2. Error Probability 0.1

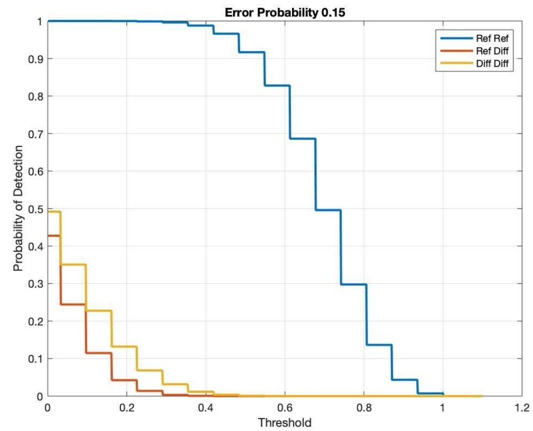


Fig. 3. Error Probability 0.15

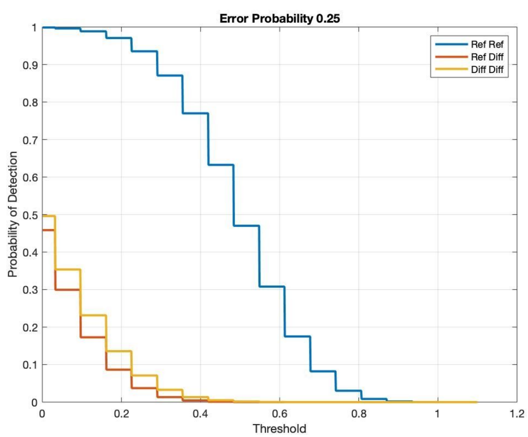


Fig. 4. Error Probability 0.25

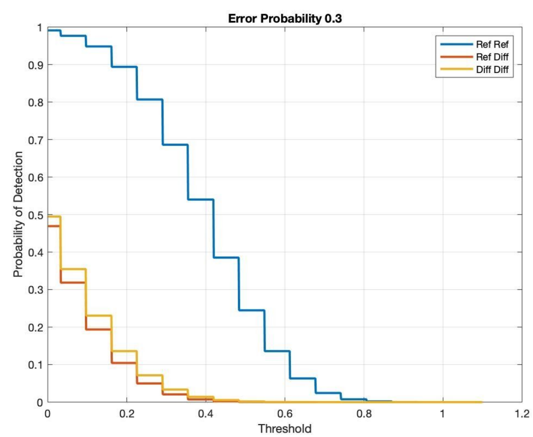
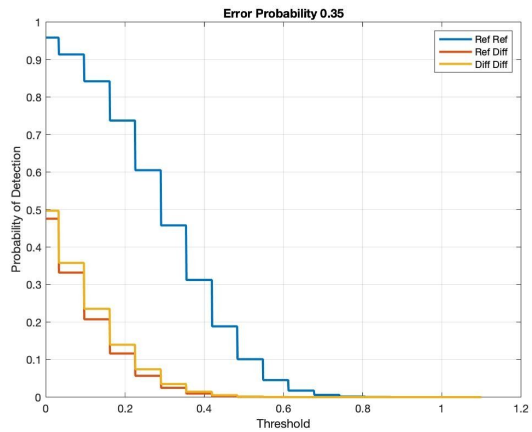


Fig. 5. Error Probability 0.3



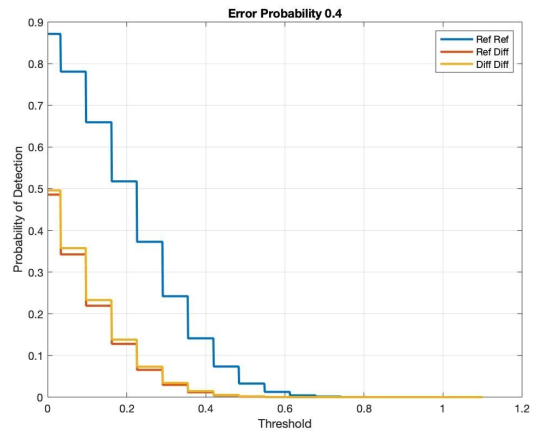


Fig. 6. Error Probability 0.35

Fig. 7. Error Probability 0.4

Fig.2 - Fig.7 as expected illustrates that as the error probability increases, the probability of detection over varying thresholds for the ref, ref case decreases. However, the ref, diff and diff, diff case stay mostly the same throughout.

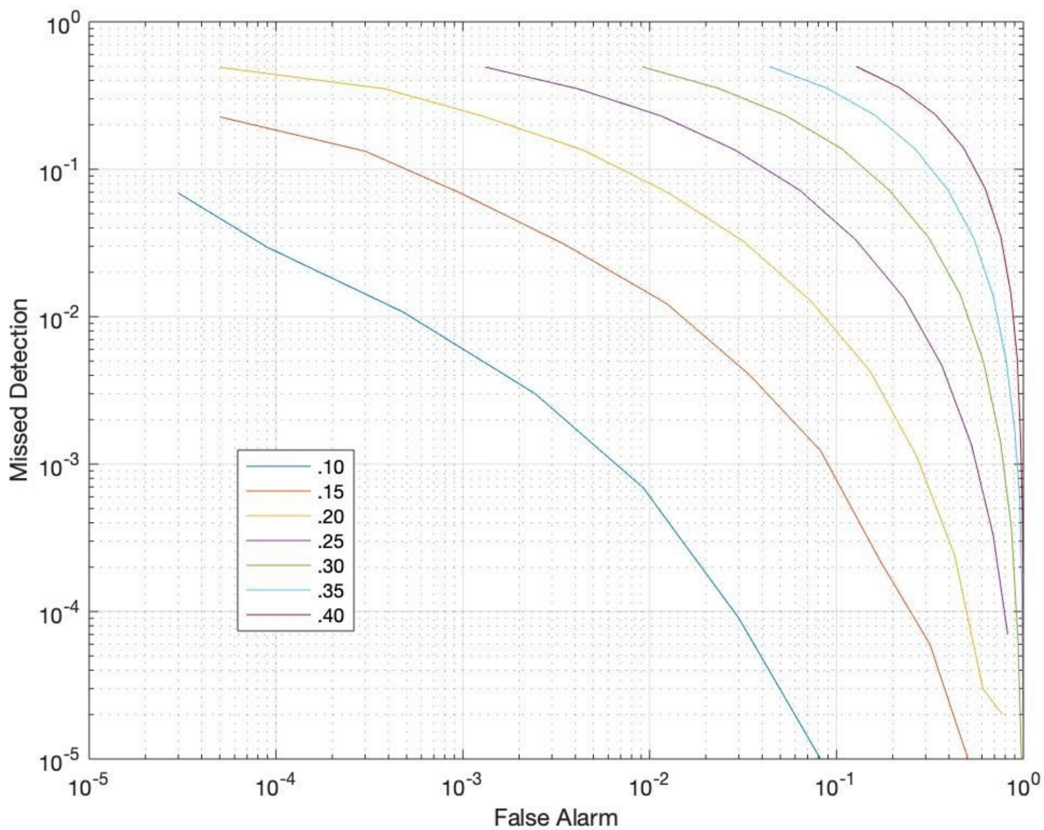


Fig. 8. NP Curve with varying error probabilities

Fig. 8 illustrates the NP curve at different values of error probability. It is seen that as the value of error probability is increasing the area of detection is decreasing, complementing the above figures.

ADDITIVE WHITE GAUSSIAN NOISE (AWGN) CHANNEL

Fig.9 - Fig. 13 illustrate the probability of detection under a range of thresholds based on the given SNR dB value. Fig. 9 shows an SNR dB value of -5, which is equivalent and close to what is used in industry and therefore used as the SNR dB to build the training dataset for the deep learning models below. It shows that for ref, ref values at a threshold value of 0 to 0.2, there is 100% probability of detection. With a threshold of 0.4 the probability of detection begins to decrease. Decreasing to 40% detection probability at a threshold of 1.2. However, for ref, diff and diff, diff sequences at a threshold value of 0 there is 0.45% and 0.5% chance of detection. With a threshold value of 0.75 having 0% detection.

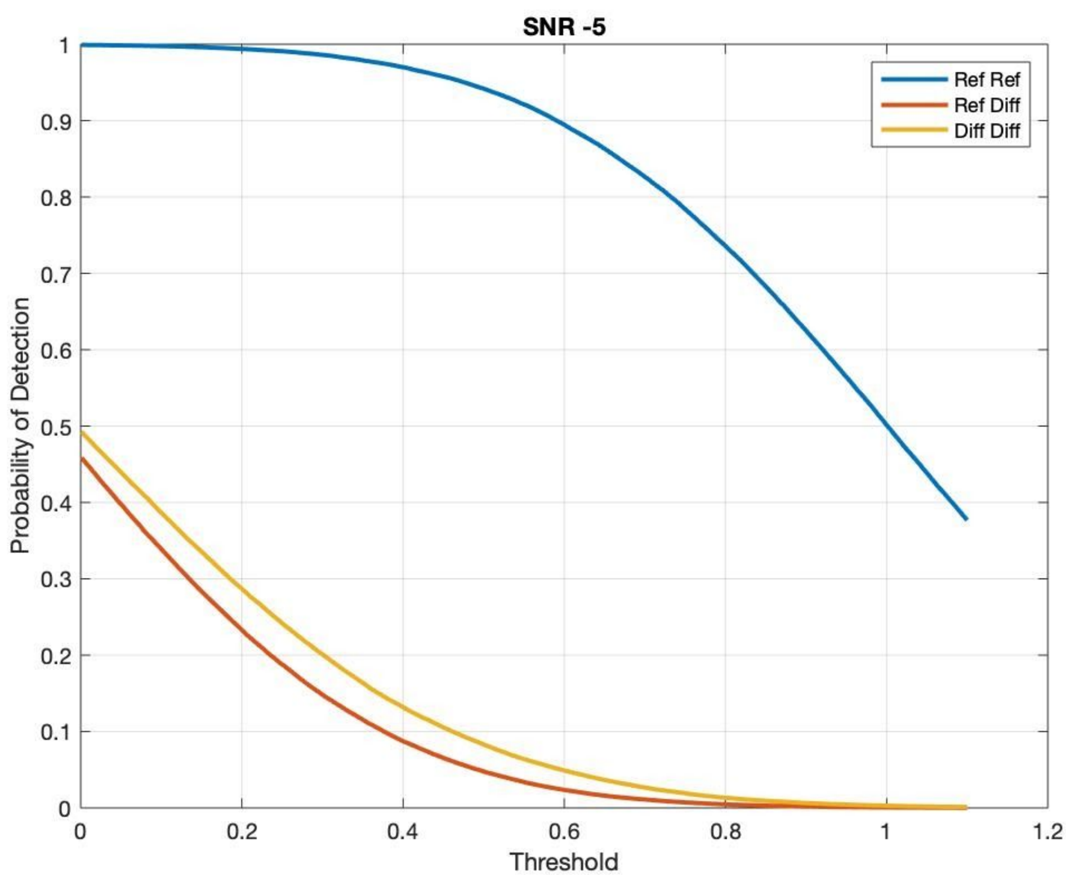


Fig. 9. SNR dB -5

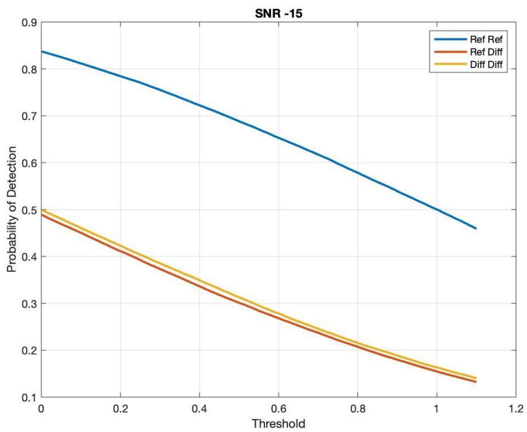


Fig. 10. SNR dB -1

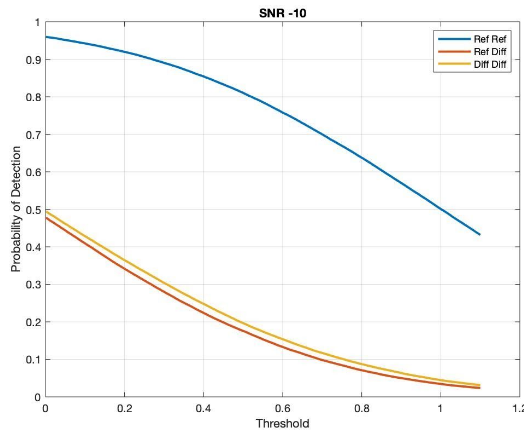


Fig. 11. SNR dB -10

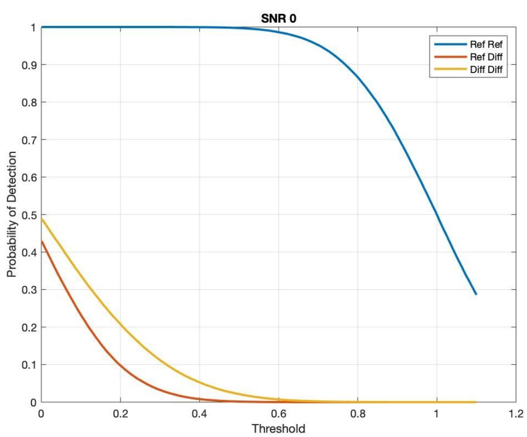


Fig. 12. SNR dB 0

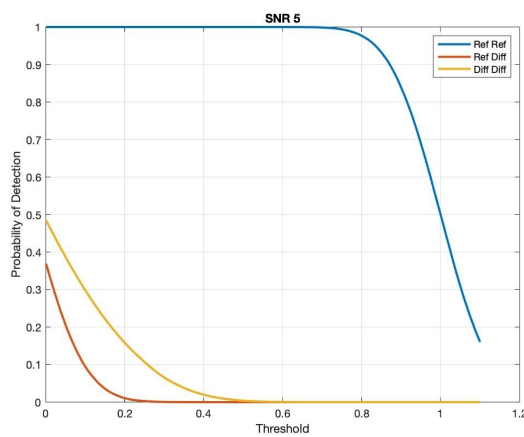


Fig. 13. SNR dB 5

As expected, as the SNR dB values increases toward one and over, the probability of detection increases for larger thresholds. As the value of SNR dB increases the less noise is in the signal. Interestingly, as the value of SNR db increases, the ref, diff and diff, diff cases decrease probability in lower thresholds. As the received signal, has multiple changes and interferences with a higher SNR dB.

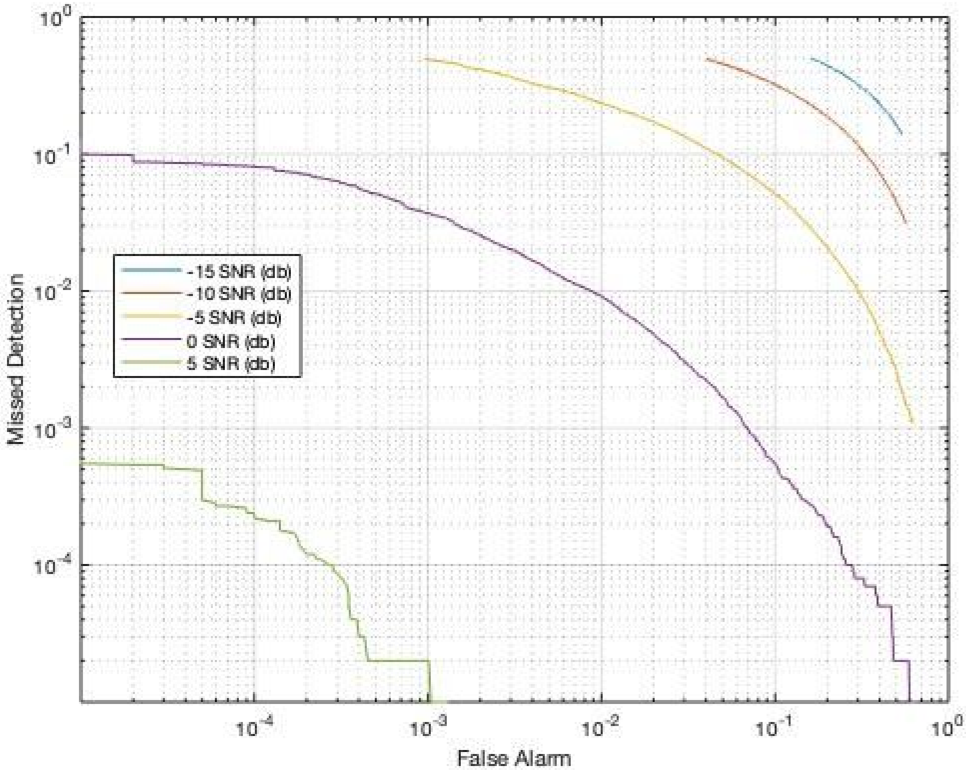


Fig. 14. NP Curve ranging from -15 SNR dB to 5 SNR -dB

Fig. 14 illustrates the NP curve at different values of SNR dB. It is seen that as the value of SNR dB increases the area of detection is increasing also, complementing the above figures. As -15 SNR dB is very high, the resulting area of detection reduces dramatically in relation to values -5 and 0 SNR dB.

MULTIPATH CHANNELS

Fig.15 - Fig. 19 illustrate the probability of detection under a range of thresholds based on multipath channels given different SNR dB values. Fig. 15 shows an SNR dB value of -5, which is equivalent and close to what is used in industry and therefore used as the SNR dB to build the training multipath dataset for the deep learning models below. It shows that for ref, ref and ref,diff values at a threshold value of 0 to 0.4, there is 100% probability of detection. However, ref, diff begins to dramatically fall in performance with ref, ref falling gradually however keeping a higher probability of detection over thresholds 0.4 to 1.2.

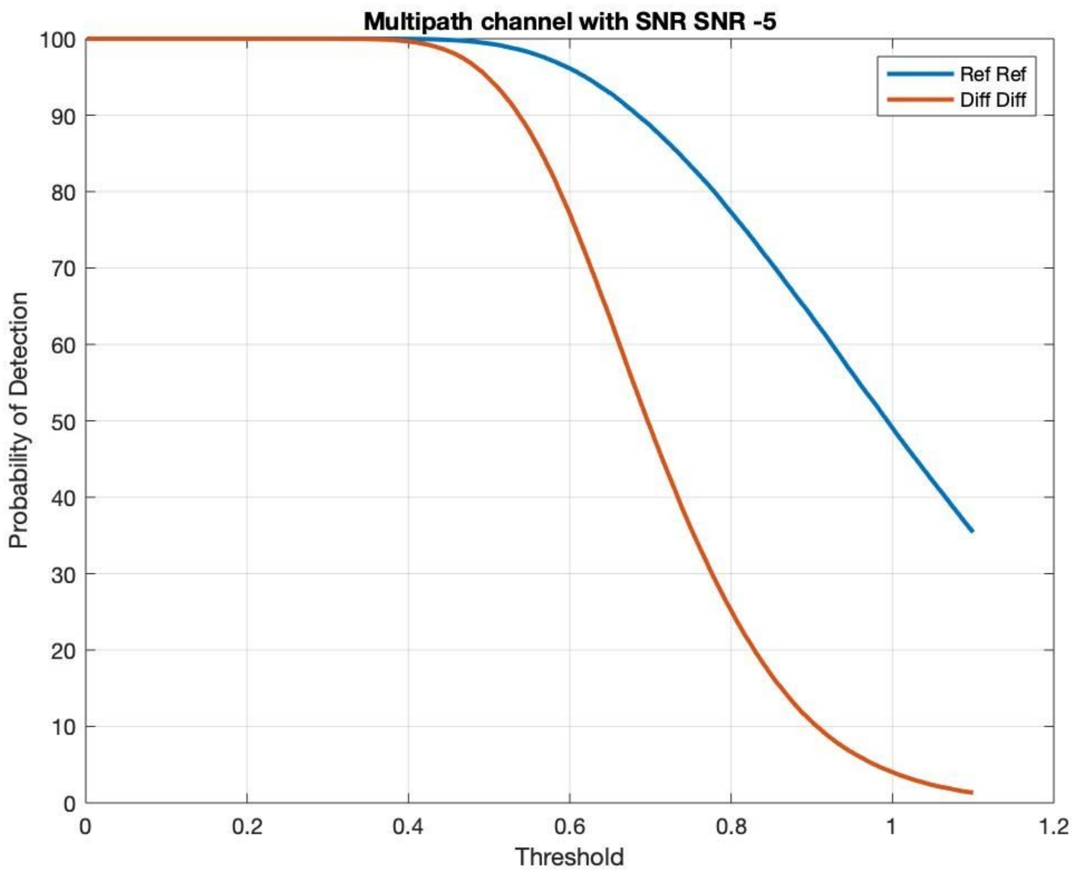


Fig. 15. Multipath Channel with SNR dB -5

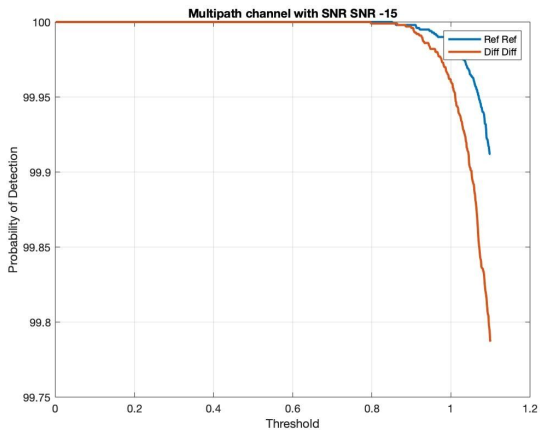


Fig. 16. Multipath SNR dB -15

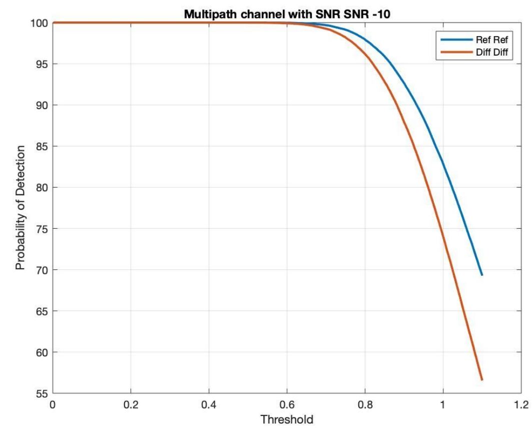


Fig. 17. Multipath SNR dB -10

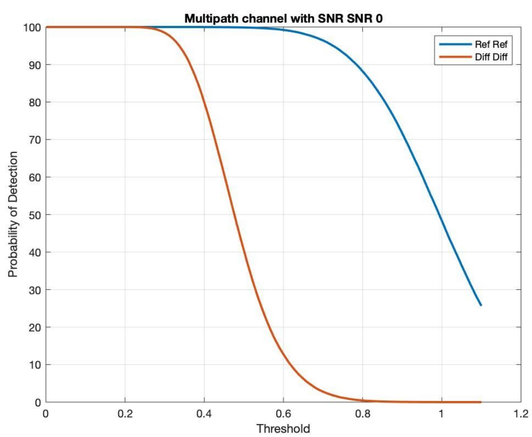


Fig. 18. Multipath SNR dB 0

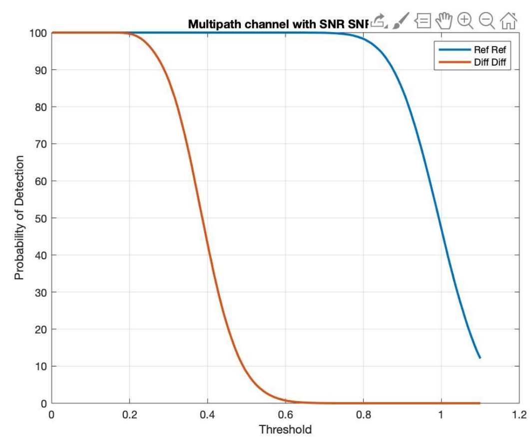


Fig. 19. Multipath SNR dB 5

As expected, as the SNR dB values increases toward one and over, the probability of detection increases for larger thresholds. As the value of SNR dB increases the less noise is in the signal. Interestingly, as the value of SNR db increases, the ref, diff cases decrease probability in lower thresholds. As the received signal, has multiple changes and interferences with a higher SNR dB.

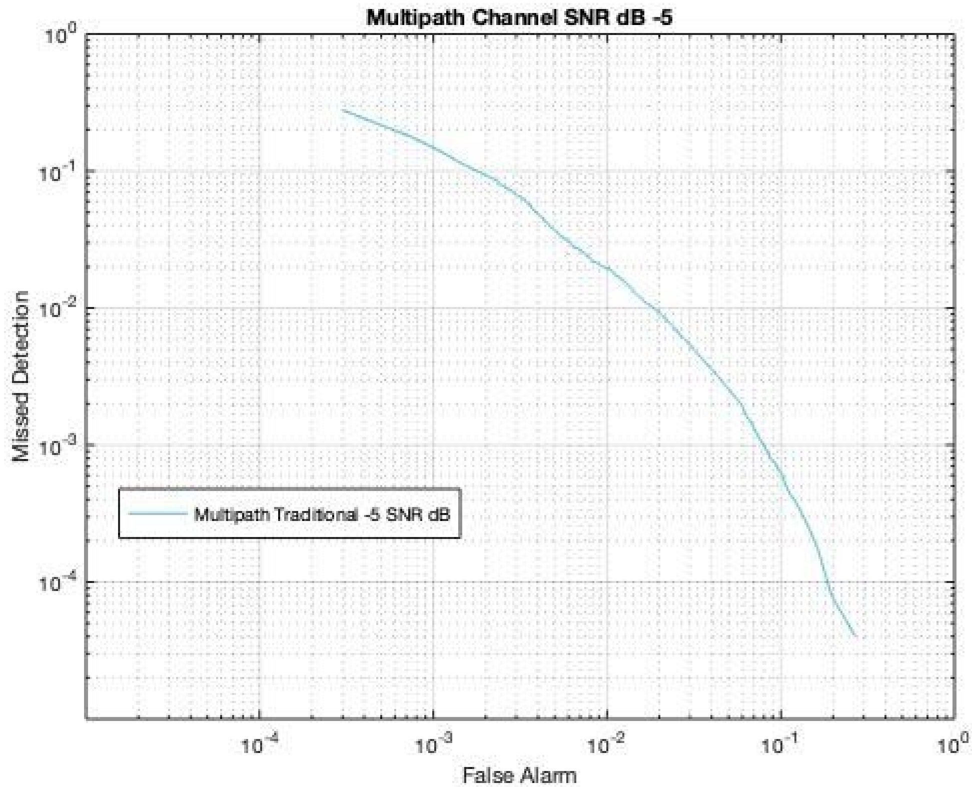


Fig. 20. NP curve for multipath channel with SNR dB -5

DEEP LEARNING PACKET DETECTION MODELS

In this section all ROC plots are shown for each architecture and each dataset, while metrics for each dataset are evaluated in the model breakdown section.

RNN

BINARY SYMMETRIC CHANNEL (BSC)

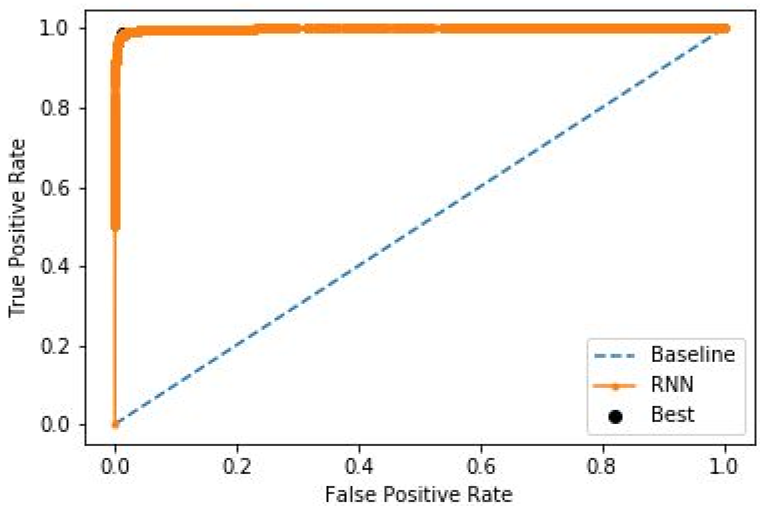


Fig. 21. BSC data RNN ROC

ADDITIVE WHITE GAUSSIAN NOISE (AWGN) CHANNEL

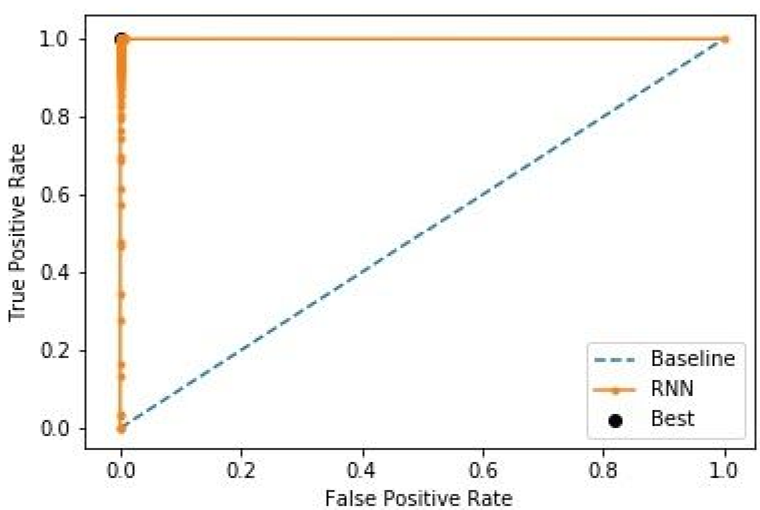


Fig. 22. AWGN data RNN ROC

MULTIPATH CHANNELS

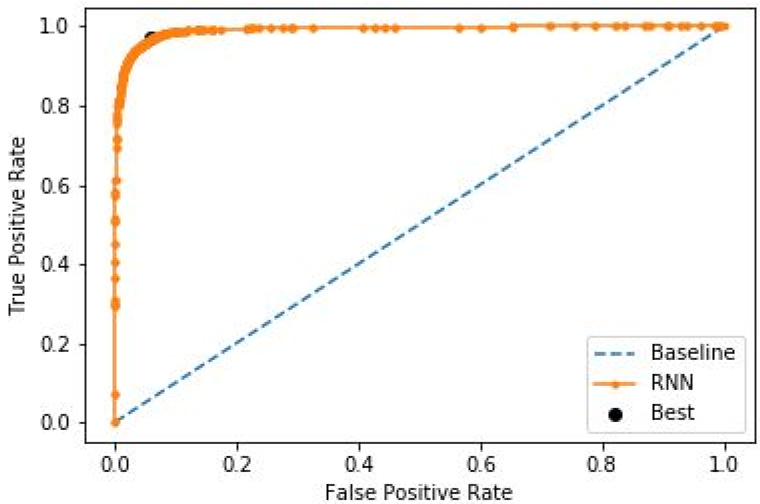


Fig. 23. Multipath data RNN ROC

RNN PERFORMANCE BREAKDOWN

The RNN architecture was trained and evaluated over three datasets as can be seen in table 1. Over all datasets the RNN performed particularly well, especially using BSC data. However this is expected because it has the least noise of the three datasets. With the addition of AWGN data however the RNN did not decrease performance too heavily with a precision of 0.95 and recall 0.92 which is a good performance over -5 SNR dB. However, interestingly the RNN model trained on multipath data outperformed the AWGN model with a precision score of 0.96 and recall of 0.94. As the multipath data is applied to real world data and applications this is a good result and has potential to be brought to a production level model.

	BSC DATA	AWGN DATA	MULTIPATH DATA
PARAMETERS	2,482	2,482	3,567
ACCURACY	0.98	0.94	0.95
PRECISION	0.97	0.95	0.96
RECALL	0.98	0.92	0.94
AUC	0.99	0.98	0.95

Table. 1. RNN Performance Evaluation

LSTM

BINARY SYMMETRIC CHANNEL (BSC)

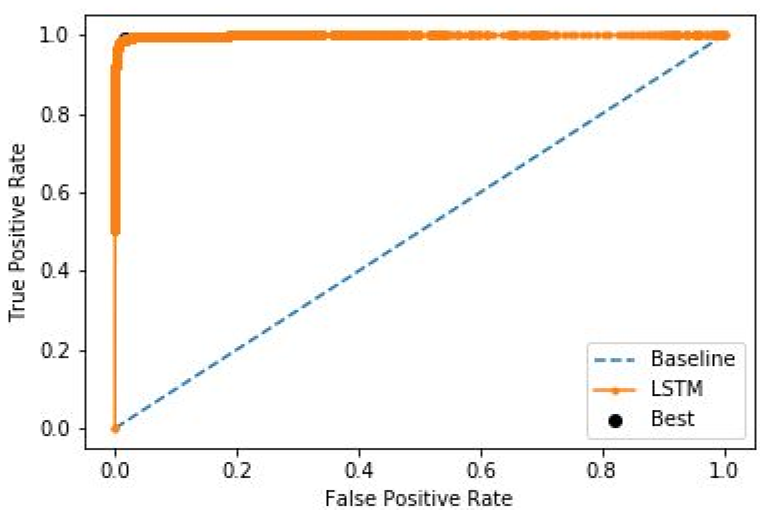


Fig. 24. BSC data LSTM ROC

ADDITIVE WHITE GAUSSIAN NOISE (AWGN) CHANNEL

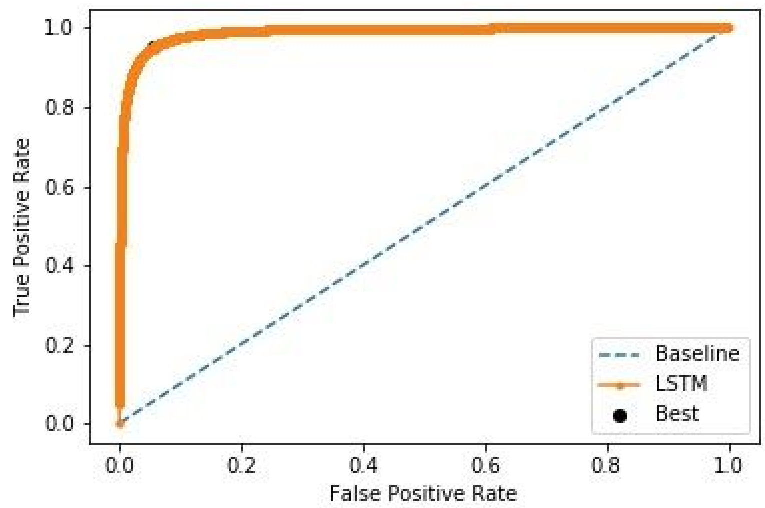


Fig. 25. AWGN data LSTM ROC

MULTIPATH CHANNELS

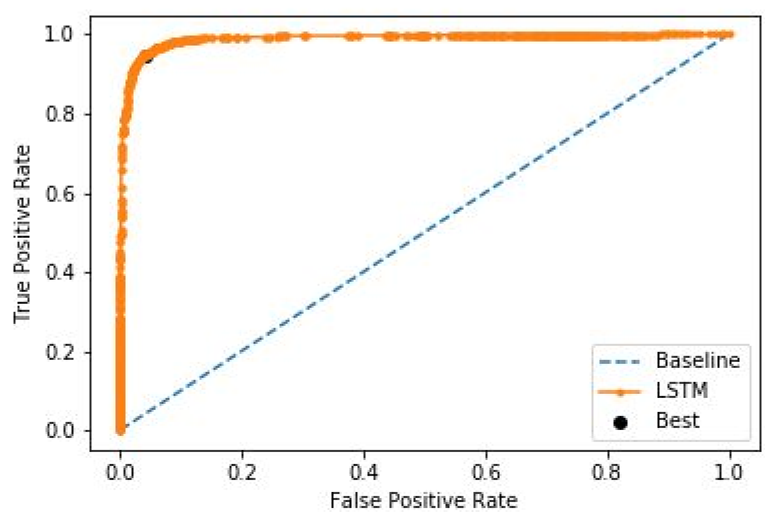


Fig. 26. Multipath data LSTM ROC

LSTM PERFORMANCE BREAKDOWN

In comparison to the RNN, the LSTM is more computationally expensive and depending on the chipset that the model will be integrated onto will determine its capabilities. Again, as expected the BSC model out performed all models comprehensively. Again, as in the RNN architecture the multipath model outperformed the AWGN model with a precision score of 0.97 in contrast to 0.94. Meaning that the multipath model correctly detected a packet 97% of the time. Again as above this model has the potential to be tested at a production level and integrated to different chipsets.

	BSC DATA	AWGN DATA	MULTIPATH DATA
PARAMETERS	2,482	15,656	23,447
ACCURACY	0.98	0.95	0.95
PRECISION	0.98	0.94	0.97
RECALL	0.98	0.94	0.94
AUC	0.99	0.98	0.95

Table. 2. LSTM Performance Evaluation

BLSTM

BINARY SYMMETRIC CHANNEL (BSC)

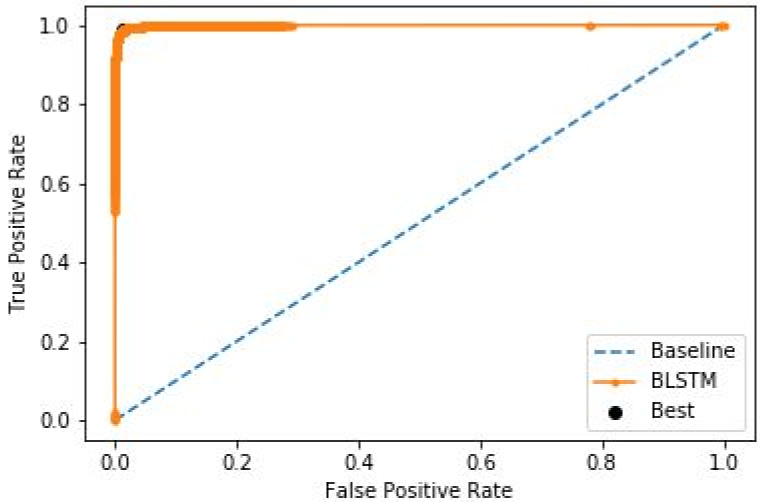


Fig. 27. BSC data BLSTM ROC

ADDITIVE WHITE GAUSSIAN NOISE (AWGN) CHANNEL

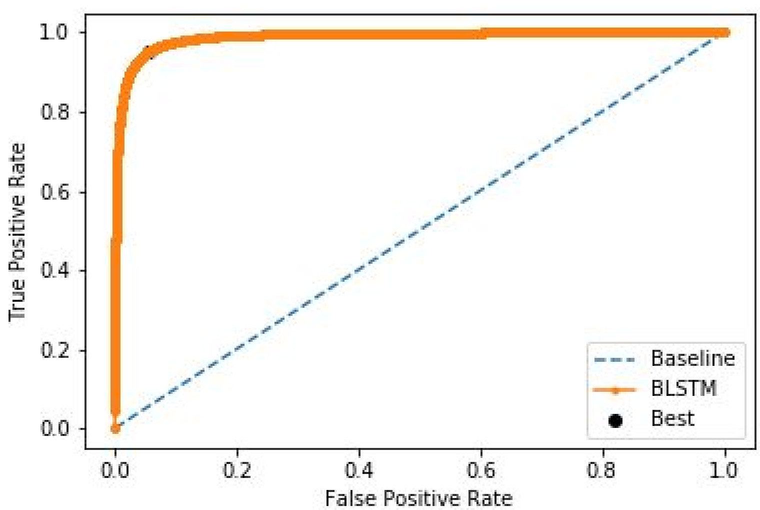


Fig. 28. AWGN data BLSTM ROC

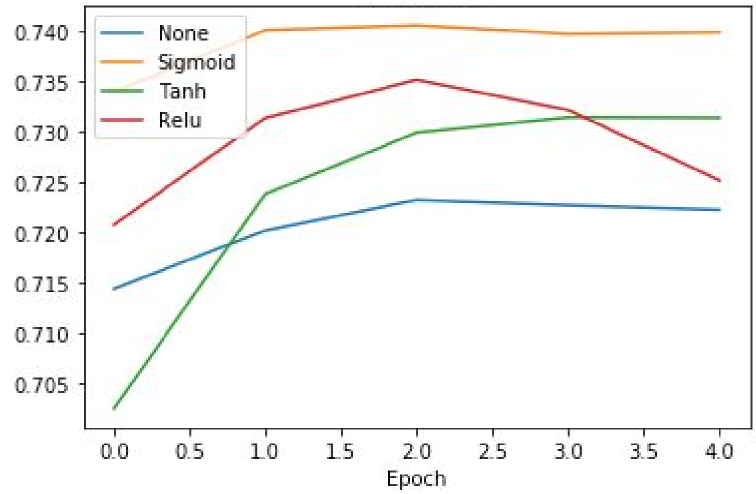


Fig. 29. Activation Function Testing

I wanted to ensure that using the sigmoid activation function was correct. I developed a never seen before test dataset separately and trained some models using the AWGN dataset over different activation functions as seen above. I found that the initial research conducted was correct and sigmoid was the correct choice to use in this case over ReLU activation.

MULTIPATH CHANNELS

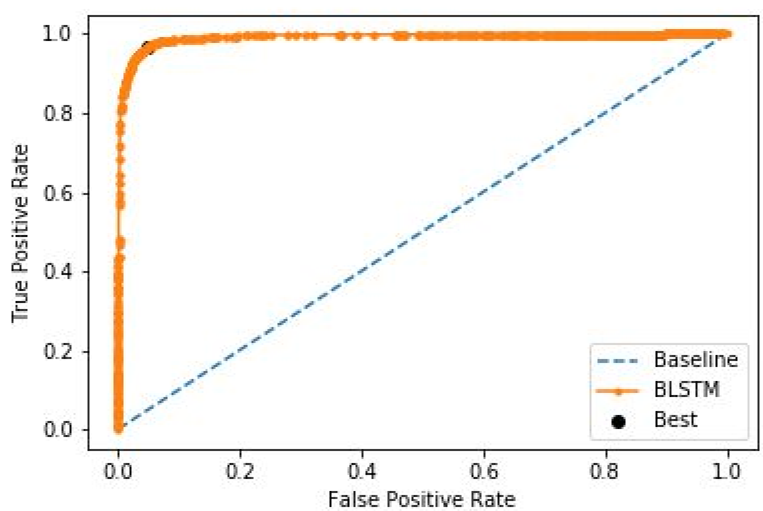


Fig. 30. Multipath data BLSTM ROC

BLSTM PERFORMANCE BREAKDOWN

As is the case with the LSTM architecture, the BLSTM architecture increases complexity as can be seen in the number of trainable parameters in each model. However, the application of BLSTM could be useful to specific chipsets in specific applications that vendors may be interested in. Again the top performing model is the BSC model as expected. Interestingly, again the multipath model outweighs the AWGN model and in precision, however has a lower recall score. However, they trade off one another resulting in largely the same detection performance.

	BSC DATA	AWGN DATA	MULTIPATH DATA
PARAMETERS	39,961	39,961	59,617
ACCURACY	0.98	0.94	0.94
PRECISION	0.98	0.95	0.97
RECALL	0.97	0.94	0.92
AUC	0.99	0.98	0.94

Table. 3. BLSTM Performance Evaluation

EVALUATION AND ANALYSIS

Fig. 31 - Fig. 33. Illustrate the NP curves between trained models and traditional methods. The objective of this project was to compare and determine the trade off between traditional packet detection methods and new state of the art deep learning approaches. Through the figures below, it can be said that deep learning approaches significantly outweigh and out perform their traditional counterparts for each dataset.

Fig.31. Sees a large increase in the performance of detection in particular the RNN and BLSTM models. However, as stated before BLSTM models carry more complexity and

varying on chipsets that these models can be integrated onto, the decision would be made on what to model to decide on.

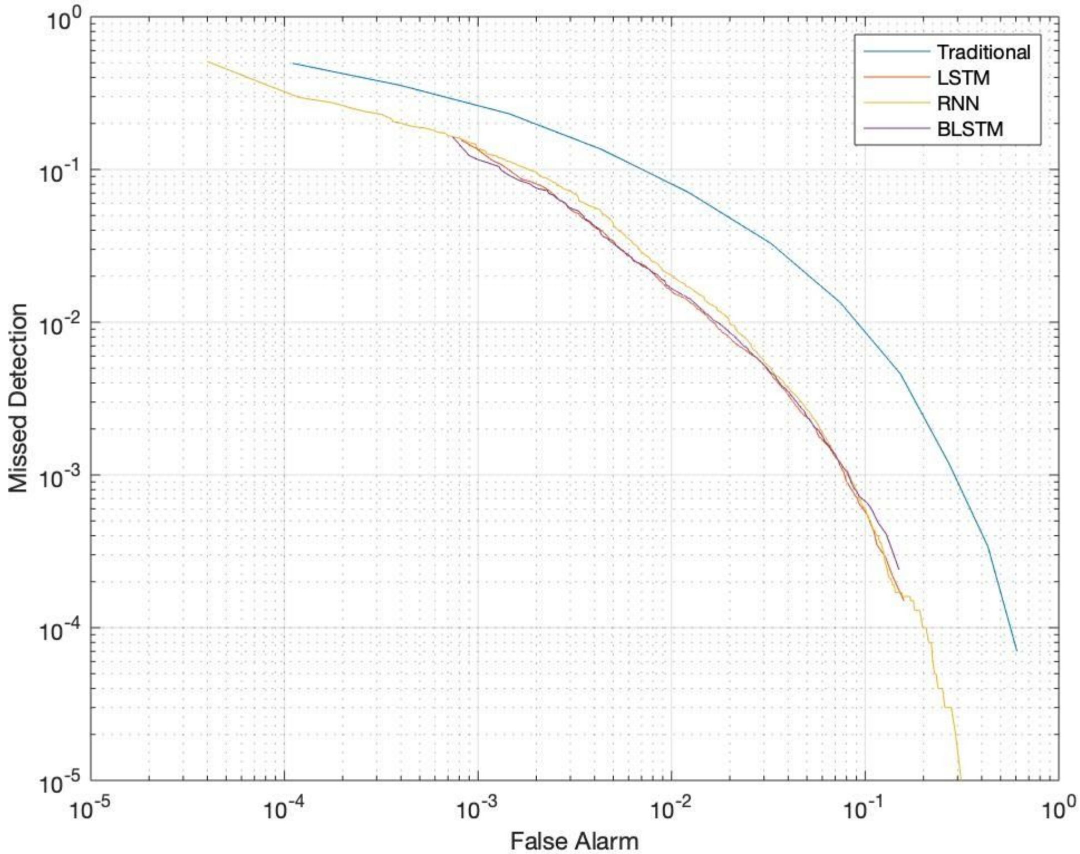


Fig. 31. NP Curves of BSC Models Vs Traditional Methods

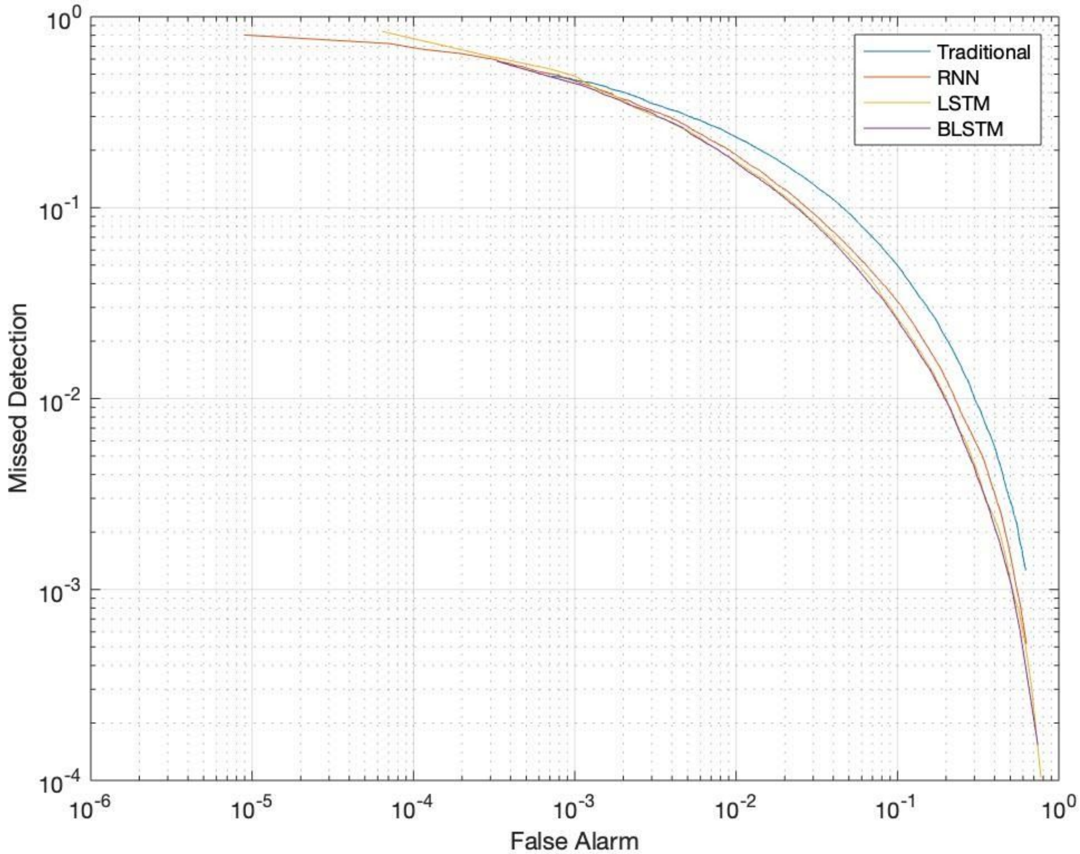


Fig. 32. NP Curves of AWGN Models Vs Traditional Methods

Fig. 32. sees a smaller increase in the performance of detection with each model performing equally illustrated by this curve. However, it can be said throughout that the AWGN models performed the worst overall.

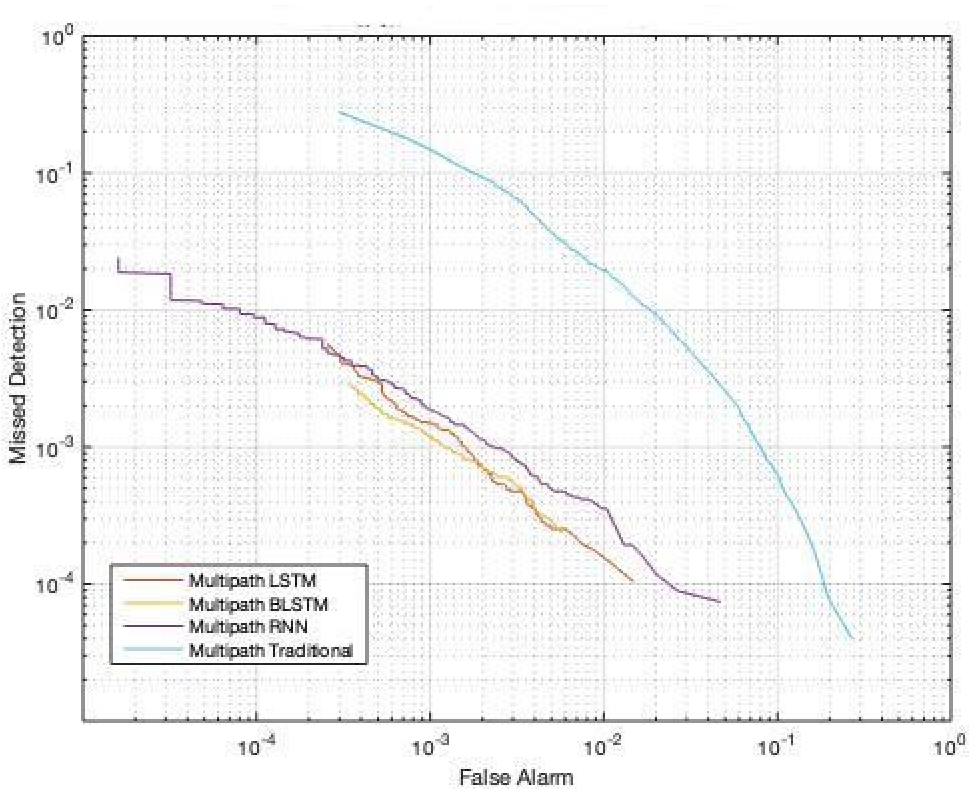


Fig. 33. NP Curves of Multipath Models Vs Traditional Methods

Fig. 33. Illustrates a large increase in the performance of detection using multipath models. This is central to real world data and the most probable models to begin integrating to chips to be tested. It can be seen here that the LSTM and BLSTM models are performing greater than the RNN however it can be seen from the metrics in table 3 that both models are quite similar. As outlined before, the computational complexity is paramount to producing the most robust and accurate model that works well under varying channel conditions.

CONCLUSION AND FUTURE PROSPECTS

In conclusion, it is evident that when comparing deep learning and traditional methods of detection that the deep learning approach significantly outperforms the traditional methods. The key aspect of this project was to find that comparison and my initial assumption that deep learning approaches would significantly increase the probability and overall packet detection in wireless networks is proven true.

The application of deep learning to this problem, will significantly reduce costs, as well as the time used redesigning chipsets. Deep learning models can be integrated through firmware, resulting in a faster, more efficient time to market.

This has proved to be encouraging research and I will continue the project with Cypress Semiconductor. Future prospects of the project include running the model on sequential data and also performing anomaly detection within packets.