

**UNIVERSITY OF KWAZULU-NATAL**

**Deep Learning-Based Space-Time  
Coding Wireless MIMO Receiver  
Optimization**

**By**

**Bhekisizwe Mthethwa**

**Supervised By: Professor Hongjun Xu**

A thesis submitted in fulfillment for the degree of  
**Doctor of Philosophy in Electronic Engineering**  
in the  
Discipline of Electrical, Electronic and Computer Engineering,  
School of Engineering,  
University of KwaZulu-Natal, Durban, South Africa

**September 2022**

## **Declaration of Authorship**

I, Bhekisizwe Mthethwa, declare that this thesis titled, 'Deep Learning-Based Space-time Coding Wireless MIMO Receiver Optimization' and the work presented in it are my own. I confirm that:

- The research reported in this thesis, except where otherwise indicated, is my original work.
- This thesis has not been submitted for any degree or examination at any other university.
- This thesis does not contain other persons data, pictures, graphs, or other information, unless specifically acknowledged as being sourced from other persons.
- This thesis does not contain other persons writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then: i) their words have been re-written, but the general information attributed to them has been referenced and ii) where their exact words have been used, their writing has been placed inside quotation marks, and referenced.
- Where I have reproduced a publication of which i am an author, co-author or editor, i have indicated in detail which part of the publication was written by myself alone and have fully referenced such publications.
- This thesis does not contain text, graphics or tables copied and pasted from the internet, unless specifically acknowledged, and the source being detailed in the thesis and in the References sections.

Signed:

Date:

As the candidate's supervisor, I, Prof Hongjun Xu, agree to the submission of this thesis.

Signed:

Date:

## **Acknowledgements**

I would like to thank my supervisor, Professor Hongjun Xu, for the academic support throughout the research. He helped me a lot to deal with the numerous publication rejections I received especially from the IEEE Journals. His technical guidance is second to none and I thank him for believing in my research abilities by generating ideas with me and letting me do the research without micromanagement. I also thank him for financially supporting me through the financial challenges I had since I had to quit my fulltime job to carry out this research. I also would like to thank the Telkom Centre of Excellence (CoE) for granting me funds to support my research. The funds assisted to keep me afloat financially. The CSIR merSETA bursary scheme needs to be acknowledged too for the financial support they awarded me to pursue my PhD degree. I am very grateful for all the formal financial support I received from these multiple financial schemes.

I would like to thank the following persons for their informal support either through financial assistance or assistance with reviewing the English grammar in my journal articles and/or PhD thesis. The names of the people are not stated in order of importance: Kgosi Tshiamo Choane (my Kgosietsile), Thandiwe Mthethwa (my mother), Nkosana Mvulane (friend), Nkosinathi Ndlela Pr. Eng (friend), Sekhotso Mofokeng (my brother), Sihle Mbathani (my good friend), Likotsi Morienyane Pr. Eng (former colleague), Sabelo Dlamini Pr. Eng (former colleague), Mihle Ngcobo (friend), Noxolo Mhlongo (friend), Qaphela Ngubane Pr. Eng (friend), Farai Mhlanga (friend), Ashton Ncube, James Mabadi Pr. Eng (friend), Mangaliso Bhengu (my good friend), Ayanda Dladla (business associate) and Sharadh Padayachee (my good client and friend).

I would like to thank Advocate Bonga Zulu, Sihle Mbathani, Dr Tladi Motsema, Kgosi Tshiamo Choane, Oyesunmisola Adejumo and Ayanda Mpama for keeping me sane with the various intellectual conversations you have engaged me on throughout my PhD degree. The PhD degree can induce depression and I thank you for keeping me sane and afloat intellectually. I would finally like to thank the almighty Horus and Kronos for

giving me strength to pull through this very challenging journey of sleepless nights and rejection. Without them I may have not met some of the good Samaritans I have acknowledged. I dedicate this PhD thesis to my late father, Jabulani Mthethwa, who instilled the spirit of intellectualism by forcing us, his children, to read Greek mythology books, reader's digest books and newspapers from a tender age. I personally value that as it ignited a spirit in me of being inquisitive.

## **Abstract**

With the high demand for high data throughput and reliable wireless links to cater for real-time or low latency mobile application services, the wireless research community has developed wireless multiple-input multiple-output (MIMO) architectures that cater to these stringent quality of service (QoS) requirements. For the case of wireless link reliability, spatial diversity in wireless MIMO architectures is used to increase the link reliability. Besides increasing link reliability using spatial diversity, space-time block coding schemes may be used to further increase the wireless link reliability by adding time diversity to the wireless link. Our research is centered around the optimization of resources used in decoding space-time block coded wireless signals. There are two categories of space-time block coding schemes namely the orthogonal and non-orthogonal space-time block codes (STBC). In our research, we concentrate on two non-orthogonal STBC schemes namely the uncoded space-time labeling diversity (USTLD) and the Golden code. These two non-orthogonal STBC schemes exhibit some advantages over the orthogonal STBC called Alamouti despite their non-linear optimal detection. Orthogonal STBC schemes have the advantage of simple linear optimal detection relative to the more complex non-linear optimal detection of non-orthogonal STBC schemes. Since our research concentrates on wireless MIMO STBC transmission, for detection to occur optimally at the receiver side of a space-time block coded wireless MIMO link, we need to optimally perform channel estimation and decoding.

USTLD has a coding gain advantage over the Alamouti STBC scheme. This implies that the USTLD can deliver higher wireless link reliability relative to the Alamouti STBC for the same spectral efficiency. Despite this advantage of the USTLD, to the best of our knowledge, the literature has concentrated on USTLD wireless transmission under the assumption that the wireless receiver has full knowledge of the wireless channel without estimation errors. We thus perform research of the USTLD wireless MIMO transmission with imperfect channel estimation. The traditional least-squares (LS) and minimum mean squared error (MMSE) used in literature, for imperfect pilot-assisted channel estimation, require the full knowledge of the transmitted pilot symbols and/or wireless

channel second order statistics which may not always be fully known. We, therefore, propose blind channel estimation facilitated by a deep learning model that makes it unnecessary to have prior knowledge of the wireless channel second order statistics, transmitted pilot symbols and/or average noise power. We also derive an optimal number of pilot symbols that maybe used for USTLD wireless MIMO channel estimation without compromising the wireless link reliability. It is shown from the Monte Carlo simulations that the error rate performance of the USTLD transmission is not compromised despite using only 20% of the required number of Zadoff-Chu sequence pilot symbols used by the traditional LS and MMSE channel estimators for both 16-QAM and 16-PSK baseband modulation.

The Golden code is a STBC scheme with spatial multiplexing gain over the Alamouti scheme. This implies that the Golden code can deliver higher spectral efficiencies for the same link reliability with the Alamouti scheme. The Alamouti scheme has been implemented in the modern wireless standards because it adds time diversity, with low decoding complexity, to wireless MIMO links. The Golden code adds time diversity and improves wireless MIMO spectral efficiency but at the cost of much higher decoding complexity relative to the Alamouti scheme. Because of the high decoding complexity, the Golden code is not widely adopted in the modern wireless standards. We, therefore, propose analytical and deep learning-based sphere-decoding algorithms to lower the number of detection floating-point operations (FLOPS) and decoding latency of the Golden code under low- and high-density M-ary quadrature amplitude modulation (M-QAM) baseband transmissions whilst maintaining the near-optimal error rate performance. The proposed sphere-decoding algorithms achieve at most 99% reduction in Golden code detection FLOPS, at low SNR, relative to the sphere-decoder with sorted detection subsets (SD-SDS) whilst maintaining the error rate performance. For the case of high-density M-QAM Golden code transmission, the proposed analytical and deep learning sphere-decoders reduce decoding latency by at most 70%, relative to the SD-SDS decoder, without diminishing the error rate performance.

# Table of Contents

Declaration of Authorship .....	ii
Acknowledgements .....	iii
Abstract.....	v
List of Figures.....	x
List of Tables .....	xii
List of Acronyms .....	xiii
Part I .....	xvii
1 Introduction.....	1
1.1 Wireless MIMO Space-time block coding .....	2
1.1.1 Orthogonal Space-time block coding.....	3
1.1.2 Non-Orthogonal Space-time block coding.....	4
1.2 Wireless MIMO Channel Estimation .....	6
1.2.1 Analytical Channel Estimation.....	6
1.2.2 Deep Learning-Based Channel Estimation .....	8
1.3 Wireless MIMO Sphere-Decoding Schemes .....	10
1.3.1 Analytical Sphere-Decoding Schemes.....	11
1.3.2 Deep Learning-Based Sphere-Decoding Schemes.....	12
2 Motivation and Research Objective.....	13
2.1 Motivation.....	13
2.2 Research Objective .....	14
3 Research Contributions .....	15
3.1 Paper A .....	16
3.2 Paper B.....	17
3.3 Paper C.....	18

4	References.....	20
Part II .....		26
Paper A .....		27
Abstract.....		28
1	Introduction.....	29
2	System Model and Channel Estimation Background .....	33
2.1	System Model .....	33
2.2	Background of Traditional Channel Estimation Methods .....	37
3	Proposed Channel Estimation For USTLD-MIMO.....	38
3.1	Proposed NN-ML Channel Estimation Hyperparameter Tuning .....	39
3.2	Training the NN-ML Channel Estimator.....	45
3.3	Optimal Transmit Power Sharing .....	47
4	Simulation Results .....	54
5	Conclusion .....	60
References .....		61
Paper B .....		65
Abstract.....		66
1	Introduction.....	67
1.1	Motivation.....	69
1.2	Contributions .....	71
2	System Model .....	73
3	Golden code SD-SDS Overview.....	75
4	Proposed Low Complexity Analytical Sphere-Decoder.....	79
5	Proposed Deep Learning-Based Sphere-Decoder.....	86
5.1	Offline Training of the DNN .....	88

5.2	Online Decoding Process .....	91
6	Simulation Results and Discussion .....	91
6.1	Complexity Analysis.....	92
6.2	Error Rate Performance .....	98
7	Conclusion .....	100
	References .....	100
	Paper C .....	103
	Abstract.....	104
1	Introduction.....	105
2	Golden Code System Model .....	110
3	Golden Code SD-SDS Overview.....	111
4	Golden Code 2-Timeslot SD-SDS Initial Radius .....	114
4.1	Derivation of Fixed Initial Radius .....	114
4.2	Adaptive Initial Radius .....	116
5	Golden Code SD-SDS Deep Learning Algorithms .....	119
5.1	SD-SDS-Radius-DNN Algorithm.....	119
5.2	SD-SDS-DNN Novel Algorithm .....	121
5.2.1	DNN Training and Testing Phase .....	123
5.2.2	SD-SDS-DNN Algorithm Explained .....	127
6	Complexity Analysis of Proposed Algorithms .....	128
7	Simulation Results and Discussions .....	133
8	Conclusion and Future Work.....	135
	References .....	136
	Part III.....	139
	Conclusion.....	140

## List of Figures

Fig. A. 1 Shows the USTLD system with blind NN-ML channel estimator with transmit power-sharing. ....	36
Fig. A. 2 Shows the equivalent noise power at 0dB SNR versus number of pilot symbols $N$ . ....	52
Fig. A. 3 Shows the power fraction versus number of pilot symbols $N$ . ....	53
Fig. A. 4 Shows the MSE performance of the $2 \times 4$ MIMO NN-ML channel estimation vs traditional channel estimation methods.....	55
Fig. A. 5 Shows the BER performance of the $2 \times 4$ MIMO NN-ML channel estimation vs traditional channel estimation methods at the same number of pilot symbols for 16-QAM modulation.....	57
Fig. A. 6 Shows the BER performance of the $2 \times 4$ MIMO NN-ML channel estimation vs traditional channel estimation methods at different number of pilot symbols for 16-QAM modulation.....	59
Fig. A. 7 Shows the BER performance of the $2 \times 4$ MIMO NN-ML channel estimation vs traditional channel estimation methods at different number of pilot symbols for 16-PSK modulation.....	60
Fig. B. 1 SD-SDS Search Tree. ....	77
Fig. B. 2 Typical DNN structure [21].....	86
Fig. B. 3 64-QAM output label distribution. ....	89
Fig. B. 4 256-QAM output label distribution. ....	90
Fig. B. 5 64-QAM Golden code SD-based detection algorithms average number of lattice points inside hypersphere for $2 \times 4$ MIMO. ....	93
Fig. B. 6 64-QAM Golden code SD-based detection algorithms average decoding latency for $2 \times 4$ MIMO.....	94
Fig. B. 7 256-QAM Golden code SD-based detection algorithms average number of lattice points inside hypersphere for $2 \times 4$ MIMO. ....	95
Fig. B. 8 256-QAM Golden code SD-based detection algorithms average decoding latency for $2 \times 4$ MIMO.....	97

Fig. B. 9 64-QAM Golden code SD-based detection algorithms error rate performance for $2 \times 4$ MIMO. ....	99
Fig. B. 10 256-QAM Golden code SD-based detection algorithms error rate performance for $2 \times 4$ MIMO. ....	99
Fig. C. 1 16-QAM skewed output label distribution for the training data. ....	125
Fig. C. 2 64-QAM skewed output label distribution for the training data. ....	125
Fig. C. 3 Complexity analysis of $2 \times 4$ MIMO 16-QAM Golden Code detection algorithms versus 256-QAM Alamouti detection algorithm.....	131
Fig. C. 4 Complexity analysis of $2 \times 4$ MIMO 64-QAM based Golden code detection algorithms versus 4096-QAM Alamouti detection algorithm.....	132
Fig. C. 5 BER for $2 \times 4$ MIMO 16-QAM Golden Code Sphere Decoding Detection Algorithms versus 256-QAM Alamouti Detection .....	134
Fig. C. 6 BER simulation results for $2 \times 4$ MIMO 64-QAM Golden Code Sphere Decoding Detection Algorithms versus 4096-QAM Alamouti Detection .....	135

## List of Tables

Table A. 1: NN-ML-Channel-Estimation Architecture and Hyper-Parameters.....	40
Table A. 2: Definition of Genetic Algorithm Parameters .....	42
Table B. 1: DNN Function Approximator Architecture.....	88
Table B. 2: 64-QAM and 256-QAM Probability thresholds .....	91
Table C. 1: DNN Radius Predictor Architecture .....	118
Table C. 2: DNN Channel Condition State Predictor Architecture.....	122
Table C. 3: 16-QAM and 64-QAM DNN Channel Condition State Predictor Parameters .....	127
Table C. 4: Estimated Transmitted Symbols Decoder Execution Percentage Distribution for 16-QAM.....	130
Table C. 5: Estimated Transmitted Symbols Decoder Execution Percentage Distribution for 64-QAM.....	132

## List of Acronyms

3 <sup>rd</sup> generation partnership project (3GPP).....	14
additive white Gaussian noise (AWGN).....	35
analog-to-digital converters (ADC).....	30
approximate linear minimum mean square error (ALMMSE).....	37
artificial neural networks (ANN).....	20
bit-error-rate (BER) .....	5
central processing unit (CPU) .....	13
channel state information (CSI).....	8
convolutional neural network (CNN).....	9
deep learning (DL).....	106
deep learning-based channel estimation and equalization scheme (DL-CE) .....	31
deep neural network (DNN).....	9
fast-essentially maximum likelihood (FML) .....	105
fast-Fourier transforms (FFT) .....	31
filter bank multicarrier	

(FBMC) .....	31
Fincke-Pohst sphere-decoder	
(FP-SD) .....	10
floating-point operations	
(FLOPS) .....	13
full-duplex	
(FD) .....	31
genetic algorithm	
(GA) .....	55
hybrid analogue-digital	
(HAD).....	9
independent and identically distributed	
(i.i.d).....	109
institute of electrical and electronic engineers	
(IEEE).....	14
internet-of-things	
(IoT) .....	15
learned denoising-based approximate message passing	
(LDAMP) .....	30
least-squares	
(LS).....	6
line of sight	
(LOS).....	109
long-term evolution	
(LTE).....	14
M-ary quadrature amplitude modulation	
(M-QAM) .....	2
maximum likelihood	
(ML) .....	2
mean squared error	
(MSE) .....	7

millimetre-wave	
(mmWave).....	8
minimum mean square error	
(MMSE) .....	6
multiple complex symbol Golden code	
(MCS-GC).....	110
multiple-input multiple-output	
(MIMO).....	1
neural network machine learning	
(NN-ML).....	28
orthogonal frequency division multiplexing	
(OFDM).....	6
quadrature amplitude modulation	
(QAM).....	29
quality of service	
(QoS).....	1
radio access network	
(RAN).....	1
radio frequency	
(RF) .....	8
random access memory	
(RAM).....	13
recurrent neural network	
(RNN).....	9
Schnorr-Euchner	
(SE).....	10
Schnorr-Euchner sphere-decoder	
(SE-SD).....	10
signal space diversity	
(SSD).....	29
signal-to-noise ratio	

(SNR) .....	5
space-time block coding	
(STBC) .....	3
sphere-decoder with detection subsets	
(SD-DS).....	11
sphere-decoder with increasing radius search	
(SD-IRS).....	10
sphere-decoder with sorted detection subsets	
(SD-SDS) .....	11
sphere-decoding	
(SD).....	65
synthetic minority over-sampling technique	
(SMOTE).....	89
time-division duplex	
(TDD) .....	7
Trellis code aided high rate space-time labeling diversity	
(TC-STLD).....	29
two-dimensional	
(2D) .....	7
uncoded space-time labelling diversity	
(USTLD) .....	5
wireless fidelity	
(WiFi) .....	14

**Part I**  
**Research Background**

# 1 Introduction

With the high demand for high spectral efficiency and highly reliable wireless message transfers between mobile end user terminals and radio access network (RAN) base stations, researchers are under pressure to create technologies that can deliver on these quality of service (QoS) requirements. One such technology that can deliver on the stringent QoS requirements is the multiple-input multiple-output (MIMO) wireless architecture. The wireless MIMO architecture is characterised by multiple transmit and receive antennas in the MIMO configuration. The wireless MIMO architecture can achieve high spectral efficiency through spatial multiplexing, via multiple transmit antennas, and deliver highly reliable wireless links via the use of multiple receive antennas [1]. There is a known trade-off between spatial multiplexing gain inherent in wireless MIMO architectures and wireless link reliability [1]. The spatial multiplexing gain is achieved by sending different data symbols on different transmit antennas, separated in space such that the transmit antennas are de-correlated, but over the same shared frequency channel. It is obvious to see that the simultaneous transmission of wireless data symbols, on the same frequency channel, causes interference which logically affects the link reliability. It is not difficult to see that as the number of transmit antennas is increased, i.e spatial multiplexing is increased, the link reliability diminishes provided the number of receive antennas is kept constant. This is, however, only true provided the wireless MIMO transmission is not encoded.

Space-time block encoding/coding (STBC) has the capability to deliver spatial multiplexing gain/full-rate and wireless link reliability/full-diversity over wireless MIMO channels. The wireless coding rate is defined, in this thesis, as the number of transmitted unique symbols per transmit antenna per transmission timeslot. Full-rate transmission is thus defined as rate-1 transmission or maximum achievable coding rate. Space-time diversity is a technique used to increase the link reliability using spatially de-correlated receive antennas and time de-correlated STBC transmission timeslots. Full diversity, in wireless STBC MIMO, is the maximum achievable diversity order of the wireless link such that the diversity order is equal to the number of STBC transmission

timeslots multiplied by the number of receive antennas. There are two types of STBC schemes namely the non-orthogonal and orthogonal STBC. The next Section 1.1 goes into detail on these two types of STBC schemes.

### ***1.1 Wireless MIMO Space-time block coding***

STBC schemes may achieve full-rate and full-diversity under specific transmit antenna MIMO configurations. The full-rate is a synonym for spatial multiplexing gain and full-diversity is a synonym for highest possible link reliability that can be delivered using the STBC and wireless MIMO configuration. The aim of the STBC schemes is to achieve simultaneous full-rate and full-diversity of a wireless MIMO system. STBC also help to achieve increased wireless link reliability in wireless MIMO configurations where the receiving device has insufficient space to install many receive antennas for high spatial diversity. In this case, the STBC add time or transmit diversity to the wireless link to compensate for the small number of receive antennas installed at the receiver.

STBC achieve time or transmit diversity by transmitting replicas of the transmitted symbols over multiple timeslots,  $T$ , and thus the time diversity is achieved via the assumption that in a different transmission timeslot, the copies of the initially transmitted M-ary quadrature amplitude modulation (M-QAM) symbols will experience independent fading. From literature it is known that, for M-QAM symbols, the orthogonal STBC cannot achieve full-rate for a wireless MIMO configuration with more than two transmit antennas [2]. This implies that the achievable capacity is undermined if the number of transmit antennas exceeds two in an open-loop wireless MIMO system using orthogonal STBC [3]. Despite the disadvantage of lack of spatial multiplexing gain from orthogonal STBC for more than two transmit antennas, they achieve full diversity whilst having linear optimal detection at the receiver due to their orthogonality property [2]. For the STBC and wireless MIMO combination to achieve full-rate and full-diversity for more than two transmit antennas, non-orthogonal STBC needs to be considered [4]. The disadvantage of non-orthogonal STBC is their non-linear optimal detection due to joint maximum likelihood (ML) detection at the receiver. The detection complexity of orthogonal STBC is polynomial with respect to the M-QAM modulation

order and that of the non-orthogonal STBC is exponential based on the dimension of number of transmit antennas and M-QAM modulation order. This implies that for a wireless MIMO system with many transmit antennas, full wireless link diversity can be achieved more efficiently using orthogonal STBC. However, for full-rate and full-diversity to be achieved for more than two transmit antennas, the more inefficient non-orthogonal STBC needs to be utilized.

### 1.1.1 Orthogonal Space-time block coding

In 1998, Alamouti developed a new simple transmit diversity scheme, for the case of two transmit antennas, that has an encoding matrix with columns that are orthogonal [5].

The Alamouti encoding matrix,  $\mathbf{A}$ , takes the form of  $\mathbf{A} = \begin{bmatrix} s_1 & -s_2^* \\ s_2 & s_1^* \end{bmatrix}$  where  $s_1$  and  $s_2$  are complex M-QAM data symbols [5]. The function  $(\cdot)^*$  denotes the complex conjugate of a complex number. The Alamouti encoding matrix has orthogonal complex column vectors, i.e.  $\begin{bmatrix} s_1 \\ s_2 \end{bmatrix}^H \begin{bmatrix} -s_2^* \\ s_1^* \end{bmatrix} = 0$ , where  $(\cdot)^H$  is the Hermitian function. This orthogonality property allows the wireless MIMO receiver to decouple, without inter-symbol interference, the simultaneously transmitted M-QAM symbols and have each M-QAM symbol independently decoded using a linear ML detector. The Alamouti STBC has the capability of achieving full diversity with linear ML optimal detection. However, the Alamouti scheme is a half-rate full-diversity scheme based on a coding rate,  $r$ , defined as  $r \triangleq \frac{n}{TN_t}$  where  $n$  is the number of transmitted unique M-QAM symbols per timeslot,  $T$  is the number of timeslots, and  $N_t$  is the number of transmit antennas. The Alamouti STBC is specifically designed for a two transmit antenna and two timeslot system. The authors in [6] and [2] were inspired by Alamouti to extend this simple transmit diversity scheme to the case of multiple transmit and receive antennas and this gave birth to orthogonal STBC. These extended orthogonal STBC schemes presented in [2] and [6] retain the property of full-diversity and simple linear ML detection from the original Alamouti transmit diversity scheme. The maximum achievable rate for the orthogonal STBC schemes presented by [2] and [6] is determined to be  $\frac{3}{4}$  for the case of three and four transmit antennas. The fact that orthogonal STBC cannot achieve full-rate or a

coding rate of 1 for more than two transmit antennas, this necessitates an investigation into non-orthogonal STBC schemes. The aim of STBC is to maximize the achievable capacity whilst enhancing the wireless link reliability. Only non-orthogonal STBC can achieve full-rate or spatial multiplexing gain and full-diversity for MIMO configurations with more than two transmit antennas [2]. The next Section 1.1.2 elaborates on the literature of non-orthogonal STBC schemes.

### *1.1.2 Non-Orthogonal Space-time block coding*

The authors in [7] propose a non-orthogonal STBC scheme which can achieve full-rate but with a loss in full-diversity. The STBC developed in [7] is called quasi-orthogonal as the encoding matrix has columns that are grouped together which are non-orthogonal to each other but orthogonal to columns in other groups. The quasi-orthogonal STBC in [7] allows pairs of transmitted M-QAM symbols to be decoded separately. Achieving full-rate at the loss of full-diversity will negatively affect throughput in mobile application services that require stringent QoS link reliability constraints to be met. In [8], the authors propose several three transmit antenna non-orthogonal STBC with a rate greater than the three transmit antenna orthogonal STBC presented in [2] and [6]. The non-orthogonal space-time block codes proposed in [8] are found using rank and determinant criterion-based code search. In [9], a full-rate full-diversity non-orthogonal space-time block code is exhibited. This STBC is called the Golden code and it has exponential optimal detection complexity of order,  $O(M^{2N_t})$ , where  $M$  is the M-QAM modulation order. The space-time block code has an encoding matrix comprised of four Golden code super symbols. The Golden code encoding matrix,  $\mathbf{B}$ , takes the following structure,  $\mathbf{B} = \begin{bmatrix} x_{11} & x_{21} \\ x_{12} & x_{22} \end{bmatrix}$ , where  $x_{11}$ ,  $x_{12}$ ,  $x_{21}$ ,  $x_{22}$  are the Golden code super symbols sent over two timeslots and two transmit antennas [9]. The Golden code space-time block code is deemed to be non-orthogonal as the encoding matrix columns are non-orthogonal, i.e.  $\begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix}^H \begin{bmatrix} x_{21} \\ x_{22} \end{bmatrix} \neq 0$ . Each Golden code super symbol conveys a linear combination of a pair of complex M-QAM data symbols.

Golden code STBC is defined in literature as a two transmit antenna and two timeslot STBC scheme. However, in [10], the authors generalise Golden code over two transmit antennas but for more than two timeslots. The Golden code super symbol also gets extended to carry multiple pairs of complex M-QAM data symbols. This generalised Golden code scheme achieves full-rate and full-diversity despite an increase in the number of timeslots. The generalised Golden code achieves higher wireless link reliability compared to the conventional Golden code scheme [10]. The authors in [11] apply space-time labelling diversity to the generalised Golden code space-time block code developed in [10] and improve the error rate performance of the generalised Golden code scheme. The new scheme, with four receive antennas and for the same spectral efficiency with the conventional Golden code, achieves a signal-to-noise ratio (SNR) gain of 1.3dB at a bit-error-rate (BER) of  $3 \times 10^{-6}$  relative to the conventional Golden code STBC [11]. Another non-orthogonal STBC is the uncoded space-time labelling diversity (USTLD) scheme which the authors in [12] propose as an alternative to the Alamouti STBC. The USTLD is shown to have coding gain over the Alamouti STBC for the same spectral efficiency. However, since the USTLD is a non-orthogonal STBC it implies that the optimal detector is a joint ML detector with an exponential detection complexity of order  $O(M^{N_t})$ . The USTLD has an encoding matrix,  $\mathbf{C}$ , which takes the form,  $\mathbf{C} = \begin{bmatrix} x_1 & \tilde{x}_2 \\ x_2 & \tilde{x}_1 \end{bmatrix}$ , where  $x_1, x_2$  are complex M-QAM symbols from the first signal constellation labelled using Gray-coding. The encoding matrix entries  $\tilde{x}_1, \tilde{x}_2$  are complex M-QAM signals, conveying the same information as  $x_1, x_2$ , but from a second signal constellation labelled using a different mapper. The second constellation mapper is optimally designed to improve the error rate performance. USTLD in [12] is designed and optimized for a two timeslot and two transmit antenna wireless MIMO system using 16-QAM and 16-PSK baseband modulation. Our research is aimed at optimizing the space-time block coded wireless MIMO receiver design and this involves optimizing the channel estimation and decoding processes. The next two Sections 1.2 and 1.3 describe the literature on the state-of-the-art of wireless MIMO channel estimation and decoding.

## ***1.2 Wireless MIMO Channel Estimation***

Channel estimation is an important component of a wireless MIMO receiver. The wireless channel needs to be known at the receiver side for efficient decoding to occur with minimal error rates. A wireless channel estimator is used in practice to estimate the wireless channel complex fading gains that a signal experiences during propagation from a transmitter to a receiver node. There are three main types of channel estimation methods namely: training-based, semi-blind and blind channel estimators [13]. The training-based channel estimators rely on pilot symbol sequences which are sent over a wireless channel for purposes of characterising the wireless channel [13]. These pilot symbols utilize expensive channel bandwidth meant for data transfer. The semi-blind channel estimators use a hybrid of training sequences or pilot symbols and the transferred data symbols to estimate the wireless channel [13]. The blind channel estimators solely use the transferred data symbols to estimate the wireless channel and do not utilize training sequences [13]. In our research, we primarily concentrate on the training-based channel estimation methods. The literature has a plethora of analytical channel estimation methods for various wireless MIMO channels, and these are illuminated in Section 1.2.1. Recently, deep learning-based channel estimators have attracted more attention in the wireless communication domain as they provide certain benefits over the known analytical methods. Section 1.2.2 deals precisely with deep learning-based wireless channel estimators.

### ***1.2.1 Analytical Channel Estimation***

The literature has a plethora of analytical channel estimators for a variety of wireless MIMO configurations. In [14], a least-squares (LS) channel estimator is exhibited that is a training-based channel estimator as it requires the use of pilot symbol sequences to derive the wireless channel estimate. The LS channel estimator requires the prior knowledge of the pilot symbol transmission matrix for it to deduce the wireless channel estimate. The LS channel estimator estimates the wireless channel using the least-squares solution equation. The LS channel estimation method is simple, but it is not as robust in mean squared error performance as the minimum mean square error (MMSE)

[15] method. The MMSE is also a training-based channel estimation method which also requires the prior knowledge of the training sequence matrix to deduce the channel estimate. The MMSE channel estimator further requires the prior knowledge of the average noise power at the receiver and the channel second order statistics [15]. These parameters may not always be known in advance and may need to be estimated which may induce more estimation errors in the wireless channel estimate.

In [16], the authors propose an efficient channel estimator for a space-time block coded orthogonal frequency division multiplexing (OFDM) wireless MIMO system. The system estimates all the necessary subcarrier channel frequency responses for the OFDM signals using a comb-type pilot symbol configuration. The channel estimator in [16] is designed to perform all its operations in the frequency domain. The computation, in frequency domain, drastically reduces the computational complexity of the channel estimator relative to the MMSE channel estimator method without loss in error rate performance. In [17], the authors discuss the LS method for MIMO-OFDM wireless systems. The mean squared error (MSE) of the LS channel estimator is derived within this context. This derived MSE is then used to determine the optimal pilot training sequences and their optimal two-dimensional (2D) placement in frequency and time. The simulations show that the optimal pilot sequences outperform the random and orthogonal pilot sequences. In [18-20], the authors propose optimal channel estimation training sequences for the wireless MIMO-OFDM configuration.

Traditional massive MIMO channel estimation suffers from high computational complexity since the wireless channel matrix is massive in dimension [21]. The conventional massive MIMO channel estimators also suffer from uplink pilot training sequence contamination and very high downlink training overhead [21]. In [22-26], the authors propose low-rank massive MIMO wireless channel estimators with some of the algorithms offering low computational complexity, all have reduced downlink training overhead which is an issue in massive MIMO channel estimation and the algorithms achieve uplink pilot decontamination. A novel semi-blind channel estimator is introduced in [27] for a cellular time-division duplex (TDD) massive MIMO wireless

configuration. This semi-blind channel estimator attenuates the interference that arises because of pilot contamination. The simulation results show that the novel semi-blind approach outperforms the traditional linear and non-linear massive MIMO channel estimators [27]. The Section 1.2.2 exhibits the recent developments in wireless MIMO channel estimation using deep learning methods. Some of the deep learning-based channel estimator methods alleviate the challenges experienced when applying the analytical channel estimators.

### ***1.2.2 Deep Learning-Based Channel Estimation***

Recently, deep learning-based channel estimators have been proposed as an alternative to analytical channel estimators. The deep-learning methods come with the advantages of being able to perform channel estimation without prior knowledge of training sequence patterns, second order channel statistics and average noise power. The traditional or analytical channel estimators usually require knowledge of channel statistics which may not necessarily always be available or known. A 2D frequency-time response deep learning-based channel estimator of a fast-fading wireless MIMO channel is presented in [28]. The paper shows that the deep learning-based channel estimator has a comparable MSE performance relative to the MMSE channel estimator. In [29], the traditional compressed sensing-based massive MIMO channel estimator is proven to be inferior relative to a deep learning-based channel estimator. In [30], the authors propose a deep learning-based frequency-time selective fading channel estimator. A deep learning-based denoising approximate message passing channel estimator is proposed in [31] that predicts the wireless channel for beam-space millimetre-wave (mmWave) massive MIMO with limited radio frequency (RF) chains. This algorithm is shown to outperform the compressed sensing-based algorithms. A deep learning channel estimator algorithm is proposed in [32] that performs channel estimation in time-selective wireless channels. This algorithm is shown to be superior in MSE performance relative to the conventional channel estimators. In [33], the authors propose a deep learning-based channel estimator that implicitly estimates the channel state information (CSI) and outperforms the conventional channel estimators when few pilot symbols are used for training. A deep learning-based channel estimator is proposed in [34] that performs

channel estimation in high-speed mobile wireless scenarios. The wireless channel being estimated experiences fast time-varying and non-stationary fading. The proposed deep learning-based channel estimator has a superior MSE performance and lower computational complexity relative to the traditional channel estimators.

A deep learning-based channel estimator for massive MIMO is proposed in [35] for the case when the pilot training sequence length is less than the number of transmit antennas in the MIMO configuration. The deep learning architecture is in the form of a multistage deep neural networks. The first stage uses a pilot training sequence for channel estimation whilst the second stage uses the transmitted data for channel estimation [35]. The second stage neural network is used to enhance the channel estimation performance. This deep learning architecture of the channel estimator produces a MSE performance that is superior to that of the traditional channel estimators for massive MIMO. In [36], the authors propose a convolutional neural network (CNN) -based autoencoder to perform channel estimation at the transmitter based on the SNR feedback from the receiver. This CNN-based autoencoder is used in a quasi-static fading wireless channel to perform joint channel estimation and pilot symbol design. For time-varying fading channels, the channel estimator is derived from a combination of a recurrent neural network (RNN) and a CNN. To address the issue of limited training samples for the deep neural network (DNN) training, generative adversarial networks are proposed in [36]. The simulation results show that the proposed DNNs have a superior performance to the state-of-the-art channel estimators under the same fading scenarios. A low training overhead deep learning-based channel estimator is proposed in [37] to learn the underlying mapping between the received omni-beam patterns and the wireless vehicular mmWave channel estimate. The MMSE channel estimator is known to be used to reduce the effect of pilot contamination in massive MIMO [38]. However, this will mean that the wireless base station will need to know the second order statistics of the channels between each user terminal and the base station which is impractical. In [38], two DNN channel estimation methods are proposed for lowering the influence of pilot contamination. The simulation results show that the two proposed DNN channel estimation methods outperform the LS and the covariance estimation methods in terms

of normalised MSE performance. A deep learning-based channel estimator is proposed in [39] that can estimate high dimensional uplink massive MIMO wireless channels in hybrid analogue-digital (HAD) transceivers. The compressed sensing algorithm is shown to be inferior relative to the deep learning-based channel estimator in terms of MSE performance and computational complexity [39]. A wireless MIMO receiver uses the estimated wireless channel to efficiently perform the transmitted message decoding process with minimal errors. Section 1.3 deals with the low complexity wireless MIMO sphere-decoders.

### *1.3 Wireless MIMO Sphere-Decoding Schemes*

Wireless MIMO optimal detection can be achieved using joint ML detection. The joint ML detector searches for complex M-QAM symbols that can minimize the objective function,  $\|\mathbf{y} - \hat{\mathbf{H}}\mathbf{u}\|_F^2$ , where  $\mathbf{y}$  is the received signal vector,  $\hat{\mathbf{H}}$  is the estimated wireless channel matrix, and  $\mathbf{u}$  is the  $N_t$ -dimensional signal vector with  $N_t$ -estimated complex M-QAM data symbols. The detection complexity of this joint ML detector is exponential in the number of transmit antennas, i.e  $O(M^{N_t})$  [40]. It is easy to see that as the number of transmit antennas becomes very large,  $N_t \rightarrow \infty$ , the detection complexity becomes extremely large even for small modulation orders since  $M \geq 4$  for M-QAM. It is thus necessary for researchers to find alternative decoding schemes that exhibit lower detection complexity relative to the joint ML detector but with marginal loss in error rate performance [40]. Complex sphere-decoding, henceforth sphere-decoding, is one such low complexity scheme with near-optimal error rate performance for wireless MIMO detection [41]. The sphere-decoder does not, on average, perform an exhaustive search for the transmitted M-QAM symbols unlike in joint ML detection. The sphere-decoder only searches for lattice points,  $\hat{\mathbf{H}}\mathbf{u}$ , that lie inside the hypersphere,  $\|\mathbf{y} - \hat{\mathbf{H}}\mathbf{u}\|_F^2 \leq r^2$ , with search radius  $r$  [41]. Sections 1.3.1 and 1.3.2 exhibit the literature on sphere-decoding of wireless MIMO systems.

### 1.3.1 Analytical Sphere-Decoding Schemes

Sphere-decoding has been a subject of interest for a long time in small and medium wireless MIMO decoding. The sphere-decoder even has variants which exhibit improvement in the dimension of detection complexity. The Schnorr-Euchner sphere-decoder (SE-SD) is a sphere-decoding variant that does not increase the search radius each time a lattice point cannot be found inside the hypersphere [42]. Instead, the Schnorr-Euchner (SE) search strategy dictates that the search must go back one layer upwards in the search tree and continue searching [42]. By not increasing the search radius and avoiding repeating the sphere-decoding search, the SE-SD exhibits lower detection complexity compared to the traditional sphere-decoder with increasing radius search (SD-IRS) found in [41]. The SE-SD has the same near-optimal error rate performance as the other sphere-decoding algorithms, for example the Fincke-Pohst sphere-decoder (FP-SD) [43], but with a reduced detection complexity. Despite the lower detection complexity compared to other sphere-decoder variants, the SE-SD still suffers from high detection complexity in large MIMO systems [44]. In [45], the authors propose a  $K$ -best sphere-decoding variant that has fixed, and low detection complexity. However, this sphere-decoding variant has a drawback of having a loss in error rate performance and exhibits high detection complexity in the high SNR region.

The application of sphere-decoding is also seen in space-time block coded wireless MIMO systems especially non-orthogonal STBC with high detection complexity like Golden code. In [46], the SE-SD is applied to Golden code detection and is shown to reduce the Golden code detection complexity without any loss in the error rate performance. It is known that the sphere-decoding detection complexity depends on the search tree search breadth and depth [47]. The search breadth is the M-QAM signal set size which must be traversed and the search depth is the number of search tree search layers which for Golden code is set to  $2N_t$ . Knowing this, in [48], the authors proposed a sphere-decoder with detection subsets (SD-DS) with the aim to reduce the search breadth of the search tree. The SD-DS is shown in [48] to have lower detection complexity relative to the SE-SD whilst maintaining the near-optimal error rate performance. In [49], a sphere-decoder with sorted detection subsets (SD-SDS) is

proposed that achieves 1 order lower detection complexity relative to the SD-DS whilst maintaining near-optimal error rate performance. The next Section 1.3.2 discusses the literature on deep learning-assisted sphere-decoding which achieves more detection complexity reduction relative to the analytical sphere-decoders.

### ***1.3.2 Deep Learning-Based Sphere-Decoding Schemes***

Deep learning in wireless MIMO decoding has recently taken centre stage in the research of wireless communication systems. New low detection complexity decoding algorithms are being developed using deep learning. In [50], a novel deep learning-based sphere-decoder, for large MIMO, is proposed that predicts the number of lattice points that lie inside a hypersphere based on an input sphere-decoding radius value. If the predicted number of lattice points that lie inside a hypersphere is large, the input radius is iteratively reduced by 50% each time until the predicted number of lattice points that lie inside the hypersphere is low. It is known that the number of lattice points found inside a hypersphere is proportional to detection complexity. The sphere-decoding radius, that is responsible for the predicted low number of lattice points that lie inside the hypersphere, is then used as an initial radius for the sphere-decoder [50]. In [51], the authors propose a low complexity deep learning-based sphere-decoder, for large MIMO, that learns to predict multiple radii which are dynamic based on the instantaneous wireless channel conditions. The deep learning-based sphere-decoder in [51] is a more efficient version of the SD-IRS in [41]. A deep learning-based sphere-decoder, for large MIMO, is proposed in [52] which predicts the minimum path metric for the subtrees of the search tree. These minimum path metrics are used to initiate the early termination of the search for candidate transmitted symbols in the search tree. The deep learning-based sphere-decoder in [52] achieves considerable reduction in detection complexity whilst maintaining near-optimal error performance. In [40], the deep learning-based sphere-decoding algorithm, for large MIMO, achieves low offline training complexity and low online decoding complexity relative to the deep learning-based sphere-decoding algorithms already available in the literature.

## 2 Motivation and Research Objective

### 2.1 Motivation

USTLD space-time block coded wireless MIMO has been studied in literature under the assumption of a wireless channel matrix that is known at the receiver without any errors. We are motivated to investigate the performance of USTLD under an imperfect wireless channel estimate. Because the signal channel bandwidth is very expensive for Telco operators, we propose a training-based channel estimation optimization technique for the USTLD-MIMO context. The literature does propose bandwidth efficient channel estimators in [53-57] but the wireless MIMO context is different. The literature in [53-56] deals with bandwidth efficient channel estimation optimization in MIMO-OFDM wireless channels. The mathematical modelling of the channel estimation bandwidth optimization objective function, in a MIMO-OFDM environment, is different to our USTLD-MIMO environment. For MIMO-OFDM, the pilot training sequence optimization is usually in 2D as the pilot symbols are located jointly in frequency and time domain whereas our USTLD-MIMO pilot symbols are located only in time-domain. In [57], the channel estimation bandwidth optimization is performed for a MIMO system with receiver frequency domain equalization. The bandwidth optimization is performed using fast Fourier and inverse Fourier transforms. Our USTLD-MIMO system does not have frequency domain equalization at the receiver hence the optimization technique in [57] is not applicable for our use case. The optimization technique for the USTLD-MIMO context entails lowering the number of time domain-based pilot training sequences necessary to perform channel estimation without losing the error rate performance. Reducing the number of pilot training sequences for the USTLD-MIMO system frees up the channel bandwidth for more data symbols to be transmitted by the users. We also propose a deep learning-based blind channel estimator that does not require the prior knowledge of the average noise power, pilot training sequence and wireless channel second order statistics unlike the traditional wireless MIMO channel estimators. The average noise power, and the wireless channel second order statistics are not always known in advance in practical wireless MIMO systems.

Golden code is a promising two transmit antenna STBC scheme for future wireless systems as it has spatial multiplexing gain over the Alamouti space-time block code. The Alamouti STBC is currently implemented in the institute of electrical and electronic engineers (IEEE) wireless fidelity (WiFi) standards [58] and the 3<sup>rd</sup> generation partnership project (3GPP) long-term evolution (LTE) standards [59]. However, despite Alamouti being implemented in modern wireless standards, it suffers from a coding rate limitation, since it is an orthogonal STBC, which limits its achievable capacity. The non-orthogonal Golden code achieves full-rate and full-diversity therefore it can deliver higher spectral efficiencies relative to the Alamouti STBC. The drawback of the Golden code is that it has exponential detection complexity whilst the Alamouti has polynomial detection complexity. This implies that the Golden code requires higher processing power and decoding latency to reliably detect transmitted symbols relative to the Alamouti scheme. This maybe one of the reasons why Golden code is not widely adopted in the modern wireless standards relative to the Alamouti STBC. We, therefore, are motivated to propose low detection complexity analytical and deep-learning-based sphere-decoders to reduce the Golden code decoding latency, for high-density M-QAM scenarios, and receiver processing power. The lower decoding latency will aid the fulfilment of the ultra-low latency required for future wireless standards whilst the Golden code spatial multiplexing or full-rate will help increase the wireless MIMO data throughputs and the full-diversity will deliver higher wireless MIMO link reliability. The Golden code reduced decoding processing power will imply that all the benefits of the Golden code STBC can be delivered to low power wireless networks such as those deployed for internet-of-things (IoT) devices.

## ***2.2 Research Objective***

Our research is aimed at optimizing the wireless receiver for non-orthogonal space-time block coded wireless MIMO systems. The wireless MIMO receiver is generally made up of a channel estimator and a message decoding algorithm. In relation to the channel estimation system, our research is focused on lowering the pilot training sequence length in a space-time block coded MIMO system since training sequences use up expensive

signal bandwidth in training-based channel estimation. We are also interested in performing blind channel estimation where being blind, in this thesis, means that the channel estimator does not need to know the pilot training sequence, average noise power and wireless channel second order statistics.

For the space-time block coded wireless MIMO decoding algorithms, we are interested in reducing the decoding latency in high-density M-QAM scenarios. The next generation wireless systems will need to deliver high data rates whilst also ensuring that the end-to-end wireless communication has ultra-low latency [60]. One way of increasing data rates, in sufficiently good quality wireless links, is by using high-density M-QAM constellations for signal transmission. These high-density M-QAM constellations deliver data via highly spectrally efficient symbols. However, the drawback of using high-density M-QAM constellations is higher modulation orders,  $M$ , that increase the decoding complexity or latency. For example, the non-orthogonal space-time Golden code has a decoding complexity that is exponential, i.e  $O(M^{2N_t})$ . It is easy to see that as we use higher-density M-QAM constellations,  $M \rightarrow \infty$ , the decoding complexity or latency increases which may aid the violation of the end-to-end ultra-low latency QoS constraint of the next generation networks. The other research objective is to lower the space-time block coded wireless MIMO decoding floating-point operations (FLOPS). Reducing the number of FLOPS needed during the decoding process implies lower receiver node random access memory (RAM) and central processing unit (CPU) loads. The lower CPU loads directly means that lower processing power is needed for decoding and thus this implies that highly reliable space-time block coded MIMO wireless links can be achieved in low power wireless communications.

### **3 Research Contributions**

The research outputs in this thesis are four published papers in which three of them are Journal articles and one is a conference paper. The publication references for these articles are listed below:

1. B. Mthethwa and H. Xu, "Deep Learning-Based Wireless Channel Estimation for MIMO Uncoded Space-Time Labeling Diversity," in *IEEE Access*, vol. 8, pp. 224608-224620, 2020. (Journal Article)
2. B. Mthethwa and H. Xu, "Golden Code Deep Learning-Aided Sphere-Decoding with Sorted Detection Subsets", SATNAC Proceedings Drakensberg, pp. 284-289, November 2021. (Conference Paper)
3. B. Mthethwa and H. Xu, "Low Complexity Golden Code Analytical and Deep Learning-Based Sphere-Decoders for High-Density M-QAM," in *IEEE Access*, vol. 10, pp. 6940-6953, 2022. (Journal Article)
4. B. Mthethwa and H. Xu, "Low Complexity Deep Learning-Assisted Golden Code Sphere-Decoding with Sorted Detection Subsets", SAIEE Africa Research Journal, vol. 113, no. 2, June 2022. (Journal Article)

The next Sections 3.1 to 3.3 elaborate on the individual contributions of each Journal article.

### **3.1 Paper A**

B. Mthethwa and H. Xu, "Deep Learning-Based Wireless Channel Estimation for MIMO Uncoded Space-Time Labeling Diversity," in *IEEE Access*, vol. 8, pp. 224608-224620, 2020.

We propose a novel bandwidth-efficient deep learning-based blind channel estimator for the USTLD-MIMO context. To the best of our knowledge, no USTLD-MIMO bandwidth-efficient deep learning-based blind channel estimator has been developed in the literature. The USTLD-MIMO system has been extensively studied in literature over wireless MIMO channels that have their channel matrix fully known at the receiver without any errors. We propose a bandwidth optimization technique that uses the transmit power-sharing between the pilot training sequences and the transmitted data symbols to achieve a reduction in bandwidth utilization. This technique is inspired by the transmit power-sharing technique used to improve the BER performance of the generalized differential scheme for spatial modulation in [61]. The transmit power-sharing scheme in [61] is used to optimize the BER performance by finding an optimal power-sharing fraction between the multiple data symbols and the single reference

symbol in the generalized differential scheme. In the USTLD-MIMO power-sharing context, the optimal power-sharing fraction is to be found between multiple transmitted data symbols and multiple pilot training sequences. The equivalent noise power [61] for the USTLD-MIMO system is derived and used to find the optimal power-sharing fraction that minimizes the MSE and BER of the USTLD-MIMO system. This optimization technique yields an 80% saving, relative to the traditional channel estimators, on the number of pilot training sequences used to perform channel estimation whilst maintaining the error rate performance for 16-QAM and 16-PSK baseband modulation.

### **3.2 Paper B**

B. Mthethwa and H. Xu, "Low Complexity Golden Code Analytical and Deep Learning-Based Sphere-Decoders for High-Density M-QAM," in *IEEE Access*, vol. 10, pp. 6940-6953, 2022.

The next generation wireless networks will need to achieve ultra-low latency, high data throughputs and very high reliability connectivity [60]. Golden code can help deliver most of these QoS requirements but its high detection complexity or decoding latency will make delivering of ultra-low latency links infeasible especially at high M-QAM modulation orders. We, therefore, propose low decoding latency analytical and deep learning-based sphere-decoders to decode high-density M-QAM Golden coded wireless MIMO messages. In proposing the analytical and deep learning-based sphere-decoders the following contributions are made:

- An instantaneous wireless channel quality metric is proposed that considers the instantaneous wireless fading power and noise power. In [46], the wireless channel quality metric depends only on the instantaneous wireless fading power. This is only valid at high SNR when fading is dominant, but at low SNR, the instantaneous noise power is dominant. Our proposed metric is valid for all SNR ranges.
- We propose instantaneously varying candidate symbol subset lengths for the sphere-decoding search tree search layers. The subset lengths vary based on the

instantaneous wireless channel quality for each transmitted M-QAM symbol encoded using Golden code super symbols. This implies that the subsets assigned to each sphere-decoding search tree search layer may have instantaneous subset lengths that differ since each M-QAM symbol experiences a different instantaneous wireless channel quality. This is different from literature where the SD-SDS [49] and SE-SD [46] Golden code detectors used fixed length candidate symbol subsets or sets.

- We then propose a worst-first search tree search order strategy that is the opposite of the best-first search tree search order strategy used in [46]. It is shown in this paper that the M-QAM symbol estimate that experiences the best instantaneous wireless channel quality has at best the smallest candidate symbol subset length assigned to it. Since the last search layer in the search tree contributes the most to the detection complexity at low SNR, assigning a subset with the smallest subset length to the last search layer of the search tree helps lower the detection complexity at low SNR.
- A deep leaning-based search tree early stopping algorithm is proposed for a small MIMO environment. In [52], the authors propose a deep learning-based search tree early termination algorithm for a large MIMO environment. The DNN architecture in [52] is however not conducive for high-density M-QAM modulation, in small MIMO, as the architecture increases proportionally in complexity as the M-QAM modulation order increases. This implies that for high data throughput Golden code applications the decoding latency will be higher. Our proposed DNN architecture is suitable for high-density M-QAM modulation as the DNN architecture complexity is invariant to the increase in M-QAM modulation order.

### 3.3 *Paper C*

B. Mthethwa and H. Xu, “Low Complexity Deep Learning-Assisted Golden Code Sphere-Decoding with Sorted Detection Subsets”, SAIEE Africa Research Journal, vol. 113, no. 2, June 2022.

Despite Golden code having spatial multiplexing gain over the Alamouti STBC, it is not widely incorporated in the modern wireless standards unlike Alamouti STBC. Golden code has a prohibitive detection complexity which limits its practical implementation in modern and future wireless standards. Since Golden code is so far restricted to a two transmit antenna MIMO configuration, it is suitable for small IoT devices that have limited antenna space. However, small IoT devices rely on battery power and therefore energy efficient wireless communication systems are paramount. A high detection complexity implies that Golden code detection requires high processing power which is not suitable for the low power wireless communications envisioned for the IoT networks. We, therefore, propose a deep learning-based low complexity sphere-decoding algorithm that lowers the Golden code detection complexity in terms of number of FLOPS. The low complexity deep learning-based wireless MIMO decoding algorithms proposed in literature, i.e [40] and [50-52], are for a large MIMO environment. We have restricted our research to a small MIMO environment and thus the complex DNN architectures used in literature will not suffice for our use case.

The proposed deep learning-based Golden code decoding algorithm achieves a comparable detection complexity to the Alamouti linear ML detector for a spectral efficiency of 8 bits/s/Hz. For a spectral efficiency of 12 bits/s/Hz, the proposed deep learning-based Golden code decoding algorithm achieves a 90% reduction in detection complexity relative to the Alamouti linear ML detector. This implies that the Golden code full-rate and full-diversity benefits can be enjoyed by the IoT low power wireless networks since the Golden code detection complexity has been reduced to at most a comparable level to that of the Alamouti linear ML detector. The proposed Golden code deep learning-based decoding algorithm also achieves a 75% reduction in detection complexity, at low SNR, relative to the SD-SDS algorithm for 16-QAM whilst maintaining the near-optimal error rate performance. For 64-QAM, the deep learning-based decoding algorithm achieves 99% lower detection complexity, at low SNR, relative to the SD-SDS algorithm whilst maintaining the near-optimal BER performance. The SD-SDS algorithm is shown in [49] to achieve lower Golden code

detection complexity relative to the traditional SD-IRS [41] and SE-SD [46] detection algorithms.

## 4 References

- [1] L. Zheng and D. N. C. Tse, "Diversity and Multiplexing: A Fundamental Trade-off in Multiple-Antenna Channels," *IEEE Trans. Inform. Theory*, vol. 49, no. 5, pp. 1073-1096, May 2003.
- [2] V. Tarokh, H. Jafarkhani and A.R. Calderbank, "Space-Time block codes from orthogonal designs," *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1456-1467, July 1999.
- [3] S. Sandhu and A. Paulraj, "Space-time block codes: a capacity perspective," *IEEE Commun. Lett.*, vol. 4, no. 12, pp. 384-386, Dec 2000.
- [4] O. Tirkkonen, A. Boariu, A. Hottinen, "Minimal non-orthogonality rate 1 space-time block code for 3+ Tx antennas," *Proc. IEEE ISSSTA2000*, vol 2, pp. 429-432, September 2000.
- [5] S. M. Alamouti, "A simple transmitter diversity scheme for wireless communications," *IEEE J. Select. Areas Commun.*, vol. 16, No. 8, pp. 1451-1458, Oct. 1998.
- [6] V. Tarokh, H. Jafarkhani, A.R. Calderbank, "The application of orthogonal designs to wireless communication," *Proc. IEEE Information Theory Workshop*, Killarney, Ireland, June 1998, pp. 46-47.
- [7] H. Jafarkhani, "A quasi-orthogonal space-time block code," *IEEE Transactions on Communications*, vol. 49, no. 1, pp. 1-4, Jan. 2001.
- [8] M. Uysal and C.N. Georghiades, "Non-orthogonal space-time block codes for 3TX antennas", *Electronics Letters*, vol. 38, no. 25, 5 Dec 2002.
- [9] J. Belfiore, G. Rekaya and E. Viterbo, "The golden code: a 2 x 2 full-rate space-time block code with non-vanishing determinants", *International Symposium on Information Theory, ISIT 2004. Proceedings.*, Chicago, IL, 2004, pp. 310-310, 2004.
- [10] H. Xu and N. Pillay, "Multiple Complex Symbol Golden Code", *IEEE Access*, vol. 8, pp. 103576-103584, 2020.

- [11] H. Xu and N. Pillay, "Multiple Complex Symbol Golden Space-Time Labeling Diversity," *IEEE Access*, vol. 9, pp. 70233-70241, 2021.
- [12] H. Xu, K. Govindasamy and N. Pillay, "Uncoded Space-Time Labeling Diversity," *IEEE Communications Letters*, vol. 20, no. 8, pp. 1511-1514, Aug. 2016.
- [13] J. K. Tugnait, Lang Tong and Z. Ding, "Single-user channel estimation and equalization," *IEEE Signal Processing Magazine*, vol. 17, no. 3, pp. 17-28, May 2000.
- [14] M. Pukkila, "Channel Estimation Modeling. Postgraduate Course in Radio-Communications", Nokia Research Center, Fall 2000.
- [15] C. R. Murthy, A. K. Jagannatham, B. D. Rao, "Training-Based and Semi-Blind Channel Estimation for MIMO Systems with Maximum Ratio Transmission", *IEEE Transactions on Signal Processing*, Vol. 54, Issue No. 7, 2006.
- [16] W.G. Jeon, K.H. Paik, and Y.S. Cho, "An efficient channel estimation technique for OFDM Systems with transmitter diversity," *IEEE Proceedings International Symposium Personal Indoor Mobile*, vol. 2, pp. 1246-50, Sept. 2000.
- [17] I. Barhumi, G. Leus, and M. Moonen, "Optimal Training Design for MIMO OFDM Systems in Mobile Wireless Channels," *IEEE Transactions on Signal Processing*, vol. 51, no. 6, pp. 1615-24, June. 2003.
- [18] B. Song, L. Gui, and W. Zhang, "Comb Type Pilot Aided Channel Estimation in OFDM Systems with Transmit Diversity," *IEEE Transactions on Broadcasting*, vol. 52, no. 1, pp. 50-7, Mar. 2006.
- [19] H. Minn, and N. Al-Dhahir, "Optimal Training Signals for MIMO OFDM Channel Estimation," *IEEE Transactions on Wireless Communications*, vol. 5, no. 5, pp. 1158-68, May. 2006.
- [20] H. Minn, N. Al-Dhahir, and Y. Li, "Optimal Training Signals for MIMO OFDM Channel Estimation in the Presence of Frequency offset and Phase Noise," *IEEE Transactions on Communications*, vol. 54, no. 10, pp. 1754-9, Oct. 2006.
- [21] H. Xie, F. Gao and S. Jin, "An Overview of Low-Rank Channel Estimation for Massive MIMO Systems," *IEEE Access*, vol. 4, pp. 7313-7321, 2016.
- [22] A. Adhikary, J. Nam, J.-Y. Ahn, and G. Caire, "Joint spatial division and multiplexing. The large-scale array regime," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6441-6463, Oct. 2013.

- [23] H. Yin, D. Gesbert, M. Filippou, and Y. Liu, "A coordinated approach to channel estimation in large-scale multiple-antenna systems," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 264-273, Feb. 2013.
- [24] X. Rao and V. K. Lau, "Distributed compressive CSIT estimation and feed-back for FDD multi-user massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3261-3271, Jun. 2014.
- [25] Z. Gao, L. Dai, Z. Wang, and S. Chen, "Spatially common sparsity based adaptive channel estimation and feedback for FDD massive MIMO," *IEEE Trans. Signal Process.*, vol. 63, no. 23, pp. 6169-6183, Dec. 2015.
- [26] H. Xie, F. Gao, S. Zhang, and S. Jin, "A unified transmission strategy for TDD/FDD massive MIMO systems with spatial basis expansion model," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3170-3184, April 2017
- [27] N. David, M. Joham, and W. Utschick. "Channel estimation in massive MIMO systems." arXiv preprint arXiv:1503.08691, Mar 2015.
- [28] M. Soltani, V. Pourahmadi, A. Mirzaei, H. Sheikhza-deh, "Deep Learning-Based Channel Estimation", *IEEE Communications Letters*, vol. 23, no. 4, pp. 652-655, April 2019.
- [29] H. He, C. Wen, S. Jin, G. Li, "Deep Learning-based Channel Estimation for Beamspace mmWave Massive MIMO Systems", *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 852-855, Oct. 2018.
- [30] Y. Yang, F. Gao, X. Ma, S. Zhang, "Deep Learning-Based Channel Estimation for Doubly Selective Fading Channels", *IEEE Access*, vol. 7, pp. 36579-36589, 2019.
- [31] H. He, C. Wen, S. Jin and G. Y. Li, "Deep Learning-Based Channel Estimation for Beamspace mmWave Massive MIMO Systems," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 852-855, Oct. 2018.
- [32] Q. Bai, J. Wang, Y. Zhang and J. Song, "Deep Learning-Based Channel Estimation Algorithm Over Time Selective Fading Channels," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 125-134, March 2020.
- [33] H. Ye, G. Y. Li and B. Juang, "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114-117, Feb. 2018.

- [34] Y. Liao, Y. Hua, X. Dai, H. Yao and X. Yang, "ChanEstNet: A Deep Learning Based Channel Estimation for High-Speed Scenarios," ICC 2019 IEEE International Conference on Communications (ICC), Shanghai, China, pp. 1-6, 2019.
- [35] C. Chun, J. Kang and I. Kim, "Deep Learning-Based Channel Estimation for Massive MIMO Systems," IEEE Wireless Communications Letters, vol. 8, no. 4, pp. 1228-1231, Aug. 2019.
- [36] J. Kang, C. Chun and I. Kim, "Deep Learning Based Channel Estimation for MIMO Systems with Received SNR Feedback," IEEE Access, vol. 8, pp. 121162-121181, 2020.
- [37] S. Moon, H. Kim and I. Hwang, "Deep learning-based channel estimation and tracking for millimeter-wave vehicular communications," Journal of Communications and Networks, vol. 22, no. 3, pp. 177-184, June 2020.
- [38] H. Hirose, T. Ohtsuki and G. Gui, "Deep Learning-Based Channel Estimation for Massive MIMO Systems with Pilot Contamination," IEEE Open Journal of Vehicular Technology, vol. 2, pp. 67-77, 2021.
- [39] J. Gao, C. Zhong, G. Y. Li and Z. Zhang, "Deep Learning based Channel Estimation for Massive MIMO with Hybrid Transceivers," IEEE Transactions on Wireless Communications, doi: 10.1109/TWC.2021.3137354.
- [40] N. T. Nguyen, K. Lee and H. DaiIEEE, "Application of Deep Learning to Sphere Decoding for Large MIMO Systems," IEEE Transactions on Wireless Communications, vol. 20, no. 10, pp. 6787-6803, Oct. 2021.
- [41] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," IEEE Transactions on Communications, vol. 51, no. 3, pp. 389-399, March 2003.
- [42] C.-P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," Mathematical programming, vol. 66, no. 1-3, pp. 181-199, 1994.
- [43] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," Mathematics of computation, vol. 44, no. 170, pp. 463-471, 1985.

- [44] N. T. Nguyen, K. Lee, and H. Dai, "QR-Decomposition-Aided Tabu Search Detection for Large MIMO Systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4857–4870, 2019.
- [45] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-best sphere decoding for MIMO detection," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 3, pp. 491–503, 2006.
- [46] L. Zhang, B. Li, T. Yuan, X. Zhang, and D. Yang, "Golden code with low complexity sphere decoder", in *Proc. 18th Int. Symp. Pers. Indoor Mobile Radio Commun.*, pp. 1-5, 2007.
- [47] J. Jaldén and B. Ottersten, "On the complexity of sphere decoding in digital communications", *IEEE Trans. Signal Process.*, vol. 53, no. 4, pp. 1474-1484, 2005.
- [48] H. Xu and N. Pillay, "Reduced Complexity Detection Schemes for Golden Code Systems", *IEEE Access*, vol. 7, pp. 139140-139149, 2019.
- [49]. H. Xu and N. Pillay, "Multiple Complex Symbol Golden Code", *IEEE Access*, vol. 8, pp. 103576-103584, 2020.
- [50] A. Askri and G. R.-B. Othman, "DNN assisted Sphere Decoder," *Int. Symp. Inf. Theory (ISIT)*, vol. 8. IEEE, Apr. 2020, pp. 70 870 – 70 877.
- [51] M. Mohammadkarimi, M. Mehrabi, M. Ardakani, and Y. Jing, "Deep Learning-Based Sphere Decoding," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4368–4378, 2019.
- [52] D. Weon and K. Lee, "Learning-Aided Deep Path Prediction for Sphere Decoding in Large MIMO Systems," *IEEE Access*, vol. 8, pp. 70870-70877, 2020.
- [53] J. P. Nair and R. V. Raja Kumar, "A bandwidth efficient channel estimation method using superimposed training for MIMO-OFDM systems," *TENCON 2008 - 2008 IEEE Region 10 Conference, Hyderabad*, pp. 1-5, 2008.
- [54] W. G. Jeon, K. H. Paik and Y. S. Cho, "An efficient channel estimation technique for OFDM systems with transmitter diversity," *11th IEEE International Symposium on Personal Indoor and Mobile Radio Communications. PIMRC 2000. Proceedings (Cat. No.00TH8525)*, London, UK, pp. 1246-1250, 2000.

- [55] C. Shin, R. W. Heath and E. J. Powers, "Blind Channel Estimation for MIMO-OFDM Systems," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 2, pp. 670-685, March 2007.
- [56] K. Josiam and D. Rajan, "Bandwidth Efficient Channel Estimation Using Super-Imposed Pilots in OFDM Systems", *IEEE Transaction on Wireless Communications*, Vol. 6, No. 6, June 2007.
- [57] J. Siew *et al.*, "A bandwidth efficient channel estimation algorithm for MIMO-SCFDE," 2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No.03CH37484), Orlando, FL, pp. 1142-1146, 2003.
- [58] F. Riera-Palou and G. Femenias, "Improving STBC Performance in IEEE 802.11n Using Group-Orthogonal Frequency Diversity", *IEEE Wireless Communications and Networking Conference*, Las Vegas, NV, pp. 193-198, 2008.
- [59] N. Sharma, "Space Time Block Code for Next Generation Multi-user MIMO System", 9th International Conference on Future Networks and Communications, *Procedia Computer Science*, pp. 172-179, 2014.
- [60] I. F. Akyildiz, A. Kak and S. Nie, "6G and Beyond: The Future of Wireless Communications Systems," *IEEE Access*, vol. 8, pp. 133995-134030, 2020.
- [61]. K. Kadathlal, H. Xu, N. Pillay, "Generalised differential scheme for spatial modulation systems", *IET Communications*, Volume: 11, Issue: 13, 2017.

**Part II**  
**Included Papers**

## **Paper A**

### **Deep Learning-Based Wireless Channel Estimation for MIMO Uncoded Space-Time Labelling Diversity**

Bhekisizwe Mthethwa and Hongjun Xu

IEEE Access, vol. 8, pp. 224608-224620,

December 2020.

## **Abstract**

Uncoded space-time labeling diversity (USTLD) is a space-time block coded (STBC) system with labeling diversity applied to it to increase wireless link reliability without compromising the spectral efficiency. USTLD achieves higher link reliability relative to the traditional Alamouti STBC system. This work aims to design a bandwidth-efficient and blind wireless channel estimator for the USTLD system. Traditional channel estimation techniques like the least-squares (LS) and the minimum mean squared error (MMSE) methods are generally inefficient in using the channel bandwidth. The LS and MMSE channel estimation schemes require the prior knowledge of transmitted pilot symbols and/or channel statistics, together with the receiver noise variance, for channel estimation. A neural network machine learning (NN-ML) channel estimator with transmit power-sharing is proposed to facilitate blind channel estimation for the USTLD system and to minimize the required channel estimation bandwidth utilization. We mathematically model the equivalent noise power and derive the optimal transmit power fraction that minimizes the channel estimation bandwidth utilization. The blind NN-ML channel estimator with transmit power-sharing is shown to utilize 20% of the LS and MMSE wireless channel estimators' bandwidth to achieve the same bit error rate (BER) performance for the USTLD system in the case of 16-QAM and 16-PSK modulation.

# 1 Introduction

Uncoded Space-Time Labeling Diversity (USTLD) is a technique developed recently by [1] to increase the link reliability of space-time block coded (STBC) systems in a multiple-input multiple-output (MIMO) environment. It uses two distinct symbol constellation mapper designs to map bitstreams to symbols. The first STBC timeslot sends information symbols from a gray coded symbol constellation mapper. The second timeslot sends the same information symbols picked from the second constellation mapper designed using the labeling technique defined in [1]. This scheme outperforms the traditional Alamouti STBC [2] system in terms of bit error rate (BER) performance as it has coding gain over the Alamouti system. In [3], the authors develop a genetic algorithm-based mapper labeling design technique for non-symmetric constellations since the USTLD mapper design in [1] is limited to symmetric constellations. In [4], the authors develop a generic analytical framework to evaluate the BER performance of USTLD in Rician, and Rayleigh fading wireless channels for a three transmit antenna MIMO system. In [5], the authors show that applying media-based modulation with radio frequency (RF) mirrors enhances the wireless link reliability of USTLD STBC schemes. The authors in [6] apply signal space diversity (SSD) to USTLD STBC in order to improve the error rate performance of USTLD. They show that the SSD USTLD scheme outperforms the USTLD BER performance. Trellis code aided high-rate space-time labeling diversity (TC-STLD) is proposed in [7] to deliver superior spectral efficiency whilst maintaining the BER performance, relative to that of USTLD. In [8], the authors investigate USTLD in a three transmit antenna MIMO configuration as the other USTLD research has only been carried out in a two transmit antenna MIMO configuration. They develop the second and third labeling mappers using a heuristic method and observe that the three transmit antenna USTLD scheme has superior BER performance relative to the two transmit USTLD scheme presented in [1]. Rectangular quadrature amplitude modulation (QAM) for USTLD is investigated in Nakagami-m fading channels in [9]. A heuristic algorithm to design the optimal labeling mapper for the rectangular QAM USTLD scheme is proposed. In [10], the authors develop a high-density M-QAM labeling mapper using a heuristic algorithm for a three transmit antenna

USTLD STBC scheme. They design the second and third labeling mappers using the heuristic algorithm for 256-QAM and 1024-QAM constellations since most research has developed mapper designs for lower modulation orders.

The works discussed so far for the USTLD scheme have assumed a perfect wireless channel estimate, which motivates the investigation of USTLD under an imperfect channel estimate. Recently, deep learning has been proposed to address challenges associated with wireless channel estimators. As stated earlier, deep learning has been proposed in the field of wireless channel estimation. In [11], the authors propose deep learning in predicting the time-frequency response of a fast-fading wireless MIMO channel. They show that the proposed deep learning algorithm has a competitive mean squared error performance relative to the traditional MMSE channel estimator. In [12], the authors prove that their proposed deep learning channel estimator outperforms the traditional compressed sensing-based algorithms for massive MIMO wireless channel estimation. The authors in [13] propose a deep learning channel estimation algorithm for doubly selective wireless fading channels. Deep learning is applied in [14] to estimate the uplink wireless channels for massive MIMO systems at the base station with some antennas using high-resolution analog-to-digital converters (ADC) and others using low-resolution ADCs. The proposed deep learning algorithm uses the high-resolution ADCs to predict the channels of the antennas using low-resolution ADCs. In [15], a deep learning-based channel estimation technique is proposed for wireless energy transfer. Based on the energy received by the energy receiver, the energy transmitter channel state information (CSI) is learned using the proposed deep learning autoencoder. The authors in [16] propose a learned denoising-based approximate message passing (LDAMP) channel estimator for beamspace millimeter-wave massive MIMO channels with limited RF chains. The deep learning-based LDAMP algorithm outperforms the compressed sensing-based algorithms. In [17], a deep learning-based channel estimator is proposed for a time-varying Rayleigh fading channel. Its mean squared error performance is shown to outperform that of the traditional channel estimation algorithms. A deep learning algorithm is proposed in [18] to handle the end-to-end wireless orthogonal frequency division multiplexing (OFDM) channels. It implicitly estimates the CSI and directly

decodes the transmitted symbols. It shows robustness relative to the conventional channel estimation techniques when fewer training pilot symbols are used. In [19], deep learning-based channel estimation and equalization scheme (DL-CE) for filter bank multicarrier (FBMC) modulation is proposed. It is shown in [19] that this DL-CE scheme achieves state-of-the-art performances in channel estimation and equalization. In [20], the authors propose a deep learning-based downlink channel estimator for fast time-varying and non-stationary wireless fading channels present in high-speed mobile scenarios. The proposed deep learning channel estimator proves to have better performance relative to the traditional channel estimators whilst offering lower computational complexity.

Research on bandwidth efficient channel estimation has been performed in literature largely in MIMO-OFDM systems. In the works presented in [21-24], the authors develop bandwidth-efficient channel estimators for the MIMO-OFDM environment. In [25], the authors propose a bandwidth-efficient channel estimator for a single carrier MIMO system with frequency domain equalization. The channel estimator iteratively uses a series of fast-Fourier transforms (FFT) and inverse FFT operations to reconstruct the CSI fully. A bandwidth-efficient blind channel estimator is proposed in [26] for a full-duplex (FD) point-to-point wireless communication system. The blind channel estimator simultaneously estimates the channel parameters of the FD system without requiring time division duplex (TDD).

In summary, it is evident from the literature that LS [27] and MMSE [28] require prior knowledge of the transmitted pilot symbols and/or the wireless channel statistics to perform channel estimation. The other general observation is that the channel estimator's mean squared error (MSE) drops as the number of transmitted pilot symbols is increased. It will be challenging to perform channel estimation using the traditional LS and MMSE channel estimators in environments where the transmitted pilot symbols and channel wireless statistics are unknown. With the high cost of licensed wireless channel bandwidth, service providers are pressured to utilize bandwidth efficiently. Hence, large numbers of pilot symbols sent over a wireless channel for channel estimation may not be desirable. Therefore, we propose a blind NN-ML channel estimator with transmit power-

sharing that minimizes the channel bandwidth usage whilst delivering a competitive MSE and BER performance compared to the traditional LS and MMSE schemes. We choose NN-ML because it does not require prior knowledge of the transmitted pilot symbols, wireless channel statistics, and receiver noise variance to perform channel estimation.

The idea of transmit power-sharing is taken from [29], where an optimal power fraction is derived that facilitates the optimal sharing of transmit power between the information symbols and a single reference symbol to improve BER performance. In our case, we apply this power-sharing technique between information symbols and multiple reference/pilot symbols to improve the MSE and BER performance of the NN-ML channel estimator relative to the traditional LS and MMSE based channel estimators. This translates to lower usage of channel bandwidth in order to deliver the same BER performance.

The main contributions of this paper are as follows:

- We propose a novel deep learning-based bandwidth-efficient blind channel estimator for the USTLD-MIMO system by employing optimal transmit power-sharing between information symbols and pilot symbols. To our knowledge, no literature has developed a bandwidth-efficient channel estimator, for the USTLD-MIMO context, using transmit power-sharing between pilot symbols and information symbols. Over and above that, the literature in [21-26] develops bandwidth-efficient MIMO channel estimators for very different system models to ours. The differences in environmental context or system model affect the method of channel estimation bandwidth optimization. We, therefore, cannot, for example, directly use a MIMO-OFDM optimized channel estimator in our system model.
- We mathematically derive a multiple pilot/reference symbol equivalent noise power upper bound for USTLD-MIMO, unlike in [29], where the equivalent noise power is only for a single reference symbol. No work in the literature has derived the equivalent noise power for the USTLD-MIMO system.
- We apply differential calculus to determine the optimal power fraction that minimizes the equivalent noise power. The minimized equivalent noise power is shown to minimize the MSE and BER of the USTLD-MIMO system. This

minimization of the MSE and BER implies a minimization of channel estimation bandwidth utilization to achieve the same BER performance as the traditional channel estimation methods.

The remainder of the paper is organized as follows: In Section 2, we present the system model for the proposed blind NN-ML channel estimator with transmit power-sharing for USTLD-MIMO and the background theory of LS and MMSE channel estimation. In Section 3, we introduce the proposed blind NN-ML channel estimator's theory with transmit power-sharing. Section 3 also presents the equivalent noise power upper bound's derivation and the optimal transmit power fraction. Section 4 discusses the MSE and BER simulation results, and Section 5 concludes the paper.

*Notation:* Bold lowercase ( $\mathbf{a}$ ) and uppercase letters ( $\mathbf{A}$ ) denote vectors and matrices, respectively.  $(\cdot)^H$  and  $\|\cdot\|_F$  are the Hermitian and the Frobenius norm of a vector or matrix, respectively.  $tr(\cdot)$  is a trace function which takes the sum total of the major diagonal of a matrix. The symbol  $\forall x$  means for all values of  $x$ . The operator  $E(\cdot)$  is the statistical expectation or mean of a random variable. The functions  $Re(\cdot)$  and  $Im(\cdot)$  return the real and imaginary components of a complex number, respectively.

## 2 System Model and Channel Estimation Background

### 2.1 System Model

A  $2 \times N_r$  USTLD system is used to evaluate the channel estimation algorithms' BER performance, where  $N_r$  is the number of receive antennas, and 2 represents the number of transmit antennas. The USTLD system is a modification of the conventional  $2 \times N_r$  Alamouti system [2]. The fundamental idea is to transmit a mapped symbol pair in the second time slot instead of the complex conjugates. The USTLD system generates the  $2 \times 2$  STBC codeword matrix based on two mappers:  $\omega_1^W$  and  $\omega_2^W$  as in [1]. For example, the two mappers for 16-QAM signal constellations are the Gray-coded labeling map  $\omega_1^{16}$  and the optimized labeling map  $\omega_2^{16}$  as per [1]. The labeling maps and their design criterion are detailed in [1]. A bitstream consisting of  $2 \log_2 W$  random bits, where  $W$  is the W-QAM/W-PSK modulation order,  $\mathbf{d} = [\mathbf{b}_{t_1}, \mathbf{b}_{t_2}]$  is fed into the

USTLD modulator to produce the modulated symbol pairs  $\mathbf{x}_1 = [x_{t_1} \ x_{t_2}]^T$  and  $\mathbf{x}_2 = [\tilde{x}_{t_2} \ \tilde{x}_{t_1}]^T$  for the first and second time slot, respectively. Let  $(\mathbf{x}_1, \mathbf{x}_2)$  be an ordered pair and let  $\mathcal{X}_{LD}$  be a set that contains all the possible modulated symbol pairs  $(\mathbf{x}_1, \mathbf{x}_2)$ , such that  $(\mathbf{x}_1, \mathbf{x}_2) \in \mathcal{X}_{LD}$ . Based on  $\mathbf{b}_{t_1}$  and  $\mathbf{b}_{t_2}$ , where  $\mathbf{b}_{t_1} = [b_1^1 \ b_2^1 \ \dots \ b_m^1]$ ,  $\mathbf{b}_{t_2} = [b_1^2 \ b_2^2 \ \dots \ b_m^2]$  with  $m = \log_2 W$ , the labeling map  $\omega_1^W$  produces  $x_{t_1}$  and  $x_{t_2}$ , while  $\omega_2^W$  produces  $\tilde{x}_{t_1}$  and  $\tilde{x}_{t_2}$ , respectively, where  $x_{t_i}$  and  $\tilde{x}_{t_i}$  belong to an W-QAM/W-PSK constellation set  $\mathcal{X}$ , with  $t_i \in [1:W]$ ,  $E\{|x_{t_i}|^2\} = E\{|\tilde{x}_{t_i}|^2\} = 1$  and  $i \in [1:2]$ .

The transmission of the  $2 \times 2$  STBC symbols happens over a quasi-static fading wireless channel with a constant channel gain over one message frame, including  $M = 200$  W-QAM/W-PSK information symbols and  $N$  channel estimation pilot symbols transmitted per frame per transmit antenna. The pilot symbols are generated using the Zadoff-Chu sequence [30] since it can generate orthogonal complex sequences of constant amplitude and varying phase. This is important to avoid creating a singular square matrix  $\mathbf{X}_r \mathbf{X}_r^H$  since the LS, and MMSE channel estimation methods rely on matrix inversion. The Zadoff-Chu sequence pilot symbols are generated using Equation (A.1)

$$P(n) = \begin{cases} e^{\frac{-j\pi Qn(n+2q)}{N}}, \forall N = 2\emptyset, \emptyset \in [1, \infty) \\ e^{\frac{-j\pi Qn(n+2q+1)}{N}}, \forall N = 2\emptyset + 1, \emptyset \in [1, \infty) \end{cases} \quad (\text{A.1})$$

where  $n \in [0:N-1]$ ,  $P(n) \in \mathbb{C}$  is the complex pilot symbol at position  $n$  of the  $N$ -dimensional pilot symbol vector,  $N$  is the number of pilot symbols transmitted per pilot symbol vector,  $j$  is a complex number,  $q \in \mathbb{N}$  and  $Q$  is a relative prime number to  $N$  and obeys the equation

$$\varphi(N, Q) = 1 \quad (\text{A.2})$$

where  $\varphi$  is the greatest common divisor function.

The wireless channel is Rayleigh frequency-flat fading. The received pilot symbols and information/message symbols at the receiver are mathematically modeled as per Equations (A.3) and (A.4):

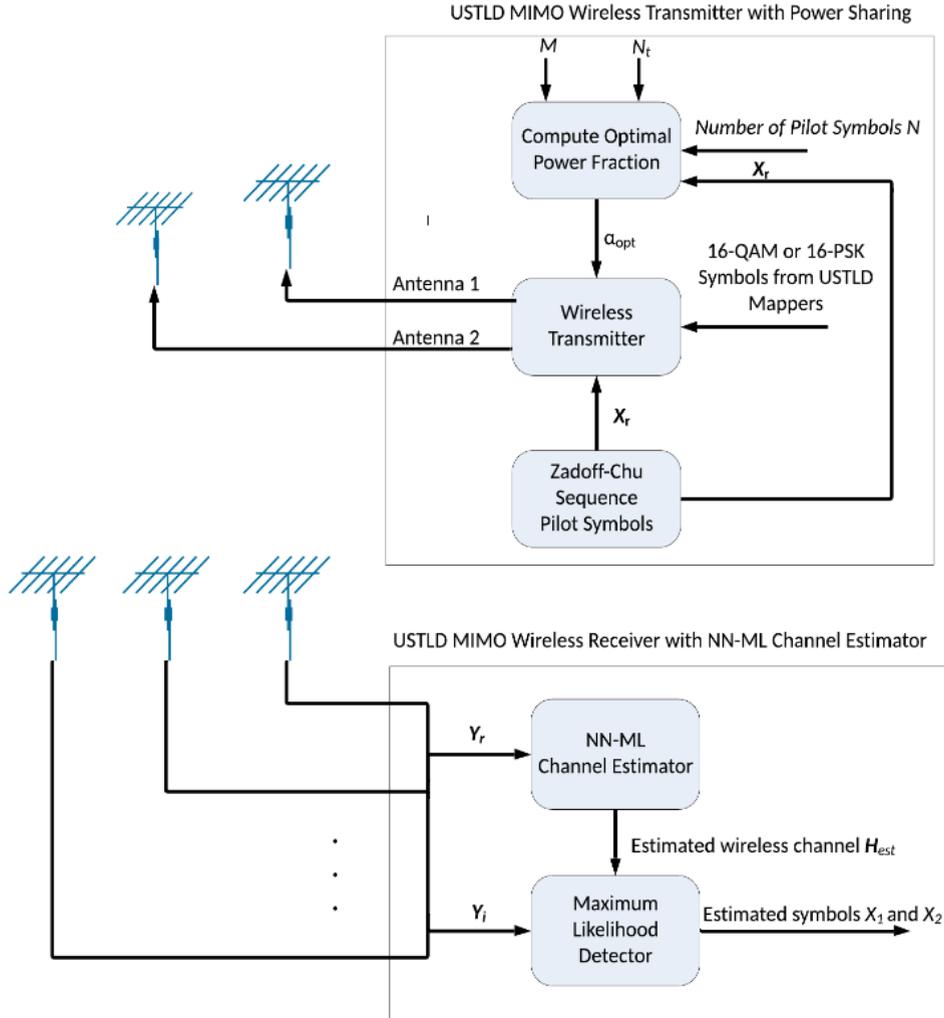
$$\mathbf{Y}_r = \mathbf{H}^T \mathbf{X}_r + \mathbf{N}_r \quad (\text{A.3})$$

$$\mathbf{Y}_i = \mathbf{H}^{T+1} \mathbf{X}_i + \mathbf{N}_i \quad (\text{A.4})$$

where  $\mathbf{H}^{T+1} = \mathbf{H}^T = \mathbf{H} \in \mathbb{C}^{N_r \times N_t}$  is the constant wireless channel matrix over one transmission frame because the wireless channel is quasi-static fading. The channel matrix  $\mathbf{H}$  has complex channel gains, which are independent and identically distributed (i.i.d) according to  $\mathbb{CN}(0,1)$ .  $\mathbf{X}_r \in \mathbb{C}^{N_t \times N}$  is the transmitted pilot symbol matrix,  $\mathbf{X}_i \in \mathbb{C}^{N_t \times 2}$  is the transmitted information symbol matrix over  $N_t = 2$ , transmit antennas, and two timeslots. The information symbol matrix  $\mathbf{X}_i$  has transmitted W-QAM/W-PSK symbols.  $\mathbf{Y}_r \in \mathbb{C}^{N_r \times N}$  is the received/observed pilot symbol matrix over  $N_r$  receive antennas,  $\mathbf{Y}_i \in \mathbb{C}^{N_r \times 2}$  is the received information symbol matrix over  $N_r$  receiver antennas and two timeslots for USTLD. The additive white Gaussian noise (AWGN) matrix  $\mathbf{N}_r \in \mathbb{C}^{N_r \times N}$  is observed at the wireless receiver over the received  $N_r \times N$  pilot symbols. The AWGN matrix  $\mathbf{N}_i \in \mathbb{C}^{N_r \times 2}$  is observed at the wireless receiver when receiving the information symbols over two timeslots. The reference/pilot noise matrix  $\mathbf{N}_r$  and the information noise matrix  $\mathbf{N}_i$  have i.i.d entries that follow the complex Gaussian distribution as follows:

$$n_i^{zx} \sim \mathbb{CN}(0, \sigma_i^2) \text{ and } n_r^{wy} \sim \mathbb{CN}(0, \sigma_r^2) \quad (\text{A.5})$$

where  $n_i^{zx}$  is the information noise matrix  $z^{\text{th}}$  row and  $x^{\text{th}}$  column entry,  $\sigma_i^2$  is the average noise power for the information receiver white noise,  $n_r^{wy}$  is the reference/pilot noise matrix  $w^{\text{th}}$  row and  $y^{\text{th}}$  column entry with an average noise power of  $\sigma_r^2$ .



**Fig. A. 1 Shows the USTLD system with blind NN-ML channel estimator with transmit power-sharing.**

As shown in Fig. A.1, the proposed system takes a fraction of the transmit power from the information symbol transmission from the wireless transmitter side and donates this transmit power fraction to the reference/pilot symbol transmission [29]. Knowing that we have  $M$  information symbols transmitted per frame per transmit antenna means we donate  $M\alpha$  transmit power to  $N$  pilot symbols. This implies that each pilot symbol gets  $\frac{M}{N}\alpha$  extra transmit power, and each information symbol loses  $\alpha$  transmit power. Mathematically this is denoted as follows:

$$\sigma_i^2 = \frac{2}{(1-\alpha)\gamma} \quad (\text{A.6.1})$$

$$\sigma_r^2 = \frac{2}{\left(1+\frac{M}{N}\alpha\right)\gamma} \quad (\text{A.6.2})$$

where  $\alpha$  is the transmit power fraction,  $M$  is the number of information symbols transmitted per frame per transmit antenna,  $N$  is the number of pilot symbols sent per frame per transmit antenna, and  $\gamma$  is the average received signal-to-noise ratio (SNR) per receive antenna. The total power for the transmission of  $M+N$  information and pilot symbols must be constant. Equations (A.6.1) and (A.6.2) obey this conservation of transmit power constraint. The optimal power fraction that ensures the optimal BER and MSE performance is computed using the  $\alpha_{opt} = f(\mathbf{X}_r, N, M, N_t)$  function that needs to be derived.

On the wireless receiver side, the NN-ML channel estimator is fed the received pilot symbol matrix  $\mathbf{Y}_r$ . The NN-ML channel estimator then predicts the wireless channel and feeds it into the maximum likelihood (ML) detector and this channel estimate is done once per received frame. The ML detector then uses the channel estimate  $\hat{\mathbf{H}}_{est}$  to detect the transmitted symbols based on the received  $\mathbf{Y}_i$  symbol matrix.

## ***2.2 Background of Traditional Channel Estimation Methods***

The LS [27] channel estimation method is the least complex channel estimation method relative to the MMSE [28] method and the approximate linear minimum mean square error (ALMMSE) [31] method but is generally the least performing of the channel estimation methods. The LS method works by generating a closed-form channel estimation formula, which estimates a wireless channel that minimizes the square of the Euclidean distance between the observed/received pilot symbol matrix and the product of the estimated wireless channel and the transmitted pilot symbol matrix. The formula for estimating the wireless channel based on the observed/received pilot vectors and the transmitted known pilot symbol matrix is as shown [27]

$$\hat{\mathbf{H}}_{LS} = \mathbf{Y}_r \mathbf{X}_r^H (\mathbf{X}_r \mathbf{X}_r^H)^{-1} \quad (\text{A.7})$$

where  $\hat{\mathbf{H}}_{LS}$  is the LS estimated wireless channel matrix,  $\mathbf{Y}_r$  is the observed/received pilot symbol matrix, and  $\mathbf{X}_r$  is the transmitted pilot symbol matrix. From Equation (A.7) it is clear that  $\mathbf{X}_r \mathbf{X}_r^H$  must be invertible and hence non-singular in nature, which motivates the selection of orthogonal pilot symbol vectors as entries in the transmitted pilot symbol matrix. Equation (A.7) also shows that the LS channel estimator requires the full knowledge of the transmitted pilot symbol matrix. The MMSE channel estimation method works by estimating the wireless channel using Equation (A.8) [28]

$$\hat{\mathbf{H}}_{mmse} = (\sigma_i^2 \mathbf{R}_{HH}^{-1} + \mathbf{X}_{mmse}^H \mathbf{X}_{mmse})^{-1} \mathbf{X}_{mmse}^H \mathbf{Y}_{mmse} \quad (\text{A.8})$$

where  $\sigma_i^2$  is the receiver noise variance,  $\mathbf{R}_{HH} = E(\mathbf{H}_{mmse} \mathbf{H}_{mmse}^H)$  is the wireless channel autocorrelation matrix,  $\mathbf{X}_{mmse} = \mathbf{X}_r^H$  is the MMSE pilot symbol matrix,  $\mathbf{Y}_{mmse} = \mathbf{X}_{mmse} \mathbf{H}_{mmse} + \mathbf{N}_{mmse}$  is the observed/received MMSE pilot symbol matrix where  $\mathbf{H}_{mmse} = \mathbf{H}^H$ . As can be seen from Equation (A.8), the MMSE channel estimator requires the full knowledge of the pilot symbol matrix, wireless channel autocorrelation statistics, and the noise variance at the receiver side. These are assumed to be known without any estimation errors.

### 3 Proposed Channel Estimation For USTLD-MIMO

The channel estimation method proposed here facilitates blind channel estimation when the transmitted pilot symbols, wireless channel second-order statistics, and the noise variance are unknown at the receiver side. The NN-ML channel estimator with transmit power-sharing method is a blind machine learning channel estimator. It also reduces the required bandwidth to achieve a good MSE and BER performance relative to the traditional channel estimation methods.

Recently, researchers within the communications research space have taken a keen interest in applying machine learning to solve communications-related research problems. The problems are primarily related to wireless symbol detectors' design using machine learning. The order of computational complexity of current expert wireless receivers is

high for higher-order modulation W-QAM/W-PSK. Machine learning comes with the benefit of training a mathematical function to predict an output based on a noisy input. Once trained, there is no need for the machine learning algorithm to search iteratively, in a large search space for higher-order modulation orders, for an estimated transmitted symbol in the case of a wireless receiver symbol detector. The function will, in a much shorter convergence time, with similar BER performance, estimate the transmitted symbol compared to a ML detector that takes a longer time to converge to a solution. This is critical for real-time communication environments as link latency needs to be minimal to achieve a good quality of service (QoS). In the case of channel estimation, we evaluate if we can train a machine-learning algorithm to predict the wireless channel based on a noisy received pilot symbol matrix with similar or better MSE and BER performance relative to the expert method of using LS or MMSE to carry out channel estimation. The advantage of using machine learning over LS or MMSE is that we can blindly estimate the wireless channel without knowing the pilot symbol matrix, channel covariance matrix, and receiver noise variance.

The work in this section is organized as follows: Section 3.1 concentrates on Hyperparameter tuning of the neural network model, Section 3.2 ventilates the training phase of the neural network model and Section 3.3 exhibits the derivation of the optimal transmit power fraction that minimizes the channel estimation bandwidth utilization.

### ***3.1 Proposed NN-ML Channel Estimation Hyperparameter Tuning***

For a supervised machine learning algorithm to be useful, it needs to be first trained using appropriate data. Two machine learning algorithms are being trained for, the first being the NN-ML-Channel-Estimation without power-sharing where  $\alpha = 0$ . The second is the NN-ML-Channel- Estimation with power-sharing, where  $\alpha = \alpha_{opt}$ . We then generated 10 000 samples for the received pilot symbols based on Equation (A.3) for  $\alpha = 0$  and  $\alpha = \alpha_{opt}$ . We also used the Zadoff-Chu sequences in Equation (A.1) to generate the transmitted pilot symbol matrix over a range of N pilot symbols. The 10 000 samples were generated for each SNR value in the range [0dB to 16dB, step 2dB]. Since this section is responsible for hyperparameter tuning, we only tuned the hyperparameters using the

10dB SNR samples in order to reduce the tuning time. Table A.1 shows the machine learning architecture and the training hyperparameters found using a genetic algorithm for the cases when  $\alpha = 0$  and  $\alpha = \alpha_{opt}$ . The hyperparameters found through genetic algorithm [32] optimization are the pseudo-random number generator seed value, learning rate and training batch size. The objective function which the genetic algorithm optimized was the validation MSE at 10dB SNR.

The NN-ML-Channel-Estimation architecture in Table A.1 was invariant to the changes in the number of pilot symbols  $N$ . However, the architecture was found to be sensitive to the MIMO receive antenna configuration  $N_r$ . This is because changing the value of  $N_r$  explicitly alters the MIMO channel matrix dimensions, which also alters the number of required neurons at the output layer of the architecture. Therefore, we declare that the architecture in Table A.1 is only valid for the  $2 \times 4$  MIMO configuration discussed in our simulation results. Any other MIMO configuration may require a new architecture and re-training of the NN-ML-Channel-Estimation model.

**Table A. 1: NN-ML-Channel-Estimation Architecture and Hyper-Parameters**

Layer	Parameter Description
Flatten	$1 \times 2N_r \times 2N$ input data dimension
(Input Layer) Dense+LeakyReLU	256 neurons. LeakyReLU slope of 0.3
BatchNormalization	void
Dropout	Probability of 0.5
(Hidden Layer) Dense+LeakyReLU	256 neurons. LeakyReLU slope of 0.3
BatchNormalization	void
Dropout	Probability of 0.5
(Output Layer) Dense+Linear	16 neurons.

Batch Size=2189,  
Learning Rate=0.00104,  
Seed Value=433

The above hyperparameters are for when  $\alpha = \alpha_{opt}$

---

Batch Size=1300,  
Learning Rate= 0.00583,  
Seed Value=549

The above hyperparameters are for when  $\alpha = 0$

The input data dimension is based on the number of transmitted pilot symbols ( $N$ ) per symbol vector and the number of receive antennas  $N_r$ . At the bottom of Table A.1 are the tunable hyperparameter values, which we derived via a genetic algorithm. The genetic algorithm pseudo-code is as follows:

**Algorithm A.1: Genetic Algorithm**

**Result:** Gene sequence with pseudo-random seed-value, batch size and learning rate for the population member that produces the lowest validation MSE at 10dB SNR.

**Initialization:**

$$\begin{aligned}
N_r &= \epsilon, \\
N &= \delta, \\
\mu_l &= 0.35 \\
\mu_g &= 0.15 \\
P &= 10 \\
\omega[1:P] &= 0 \\
G &= 100 \\
\pi &= 0.20 \\
E &= 1000
\end{aligned}$$

**Step 1:** *Generate  $P$  random population members with random hyperparameter values.*

Let each population member be denoted by  $\Omega_k$  where  $k \in [1:P]$ .  $\Omega_k$  contains the set  $\{\lambda_k, \eta_k, \zeta_k\}$  where  $\lambda_k$  is the pseudo-random seed value,  $\eta_k$  is the learning rate and  $\zeta_k$  is the batch size for the  $k^{th}$  population member. The random values are set within the following constraints:  $\lambda_k \in [2, 1001) \subseteq \mathbb{Z}$ ,  $\eta_k \in [0.0001, 0.01] \subseteq \mathbb{R}$ , and  $\zeta_k \in [1000, 4000] \subseteq \mathbb{Z}$ .

**Step 2:** *Set  $i = 1$*

**Step 3:** *IF  $i \leq G$  THEN*

- a) *Set  $k = 1$*
- b) *IF  $k \leq P$  THEN*
  - I.  *$parent_k = \Omega_k$*
  - II.  *$\lambda_k = parent_k.getSeedValue()$*
  - III.  *$\eta_k = parent_k.getLearnRate()$*
  - IV.  *$\zeta_k = parent_k.getBatchSize()$*
  - V.  *$\chi = Import(Dataset)$*
  - VI.  *$\omega[k] = \Phi(\chi, \pi, E, \lambda_k, \eta_k, \zeta_k, N_r, N)$*
  - VII.  *$k = k + 1$*

VIII. GOTO Step 3b

c) ELSE

I. Set  $seq = 1$

II. IF  $seq \leq P$  THEN

- $\tau[1:P] = \text{getFitnessValues}(\omega[1:P])$
- $\rho[1:P] = \text{getSelectionProbabilities}(\tau[1:P])$
- $v[1:2] = \text{selectParents}(\rho[1:P])$
- $c = \text{produceChild}(v[1:2])$
- $\Omega_{seq} = c$
- $seq = seq + 1$
- GOTO Step 3c number II.

III.  $i = i + 1$

IV. GOTO Step 3

**Step 4: End**

The initialization step initializes the salient variables defined in Table A.2, amongst other variables in Algorithm A.1.

**Table A. 2: Definition of Genetic Algorithm Parameters**

Parameter	Description
$N_r$	number of receive antennas in MIMO configuration
$N$	Number of pilot symbols
$\mu_l$	Local mutation rate
$\mu_g$	Global mutation rate
$P$	Population size per generation
$\omega[1:P]$	A list that stores the MSE validation loss values at 10dB SNR for each $\Omega_k$ .
$G$	Maximum number of evolutionary generations
$\pi$	The fraction of the dataset which will be used as the test data.
$E$	Number of training Epochs
$\tau[1:P]$	A list that stores the fitness value for each $\Omega_k$ .
$\rho[1:P]$	A list that stores the selection probabilities for each $\Omega_k$ .
$v[1:2]$	A list that stores the two parents that are selected for mating
$c$	The child produced from the mating of the two selected parents.
$\chi$	Dataset

Algorithm A.1 works by generating  $P$  random population members with randomized seed-values, batch sizes and learning rates as an initial Genetic algorithm population of possible hyperparameters. The population members,  $\Omega_k$ , hyperparameters are randomly assigned values subject to the constraints stated in Step 1 of Algorithm A.1. The algorithm then iterates through each population member per generation. It uses each population member's hyperparameters to train the neural network architecture in Table A.1, represented as  $\Phi(\cdot)$  in the Algorithm. The neural network architecture is trained with a training dataset of 80% of the imported dataset stored in  $\chi$ . The other 20% is used as a test dataset to produce the validation MSE at 10dB SNR. The dataset stored in  $\chi$  is produced from Equation (A.3), at 10dB SNR, for the power fraction values  $\alpha = 0$  and  $\alpha = \alpha_{opt}$ . The training dataset is collected at a single SNR value of 10dB because at this stage, we are merely pre-training the neural network to select hyperparameters that produce the lowest validation MSE at 10dB SNR. At this stage, we are not fully training the neural network for channel estimation but for tuning or selecting optimal hyperparameters.

The population members in the current generation are selected one after the other, and their hyperparameters, together with the training and test datasets, are used to train and test the neural network in Table A.1 in order to evaluate the validation MSE at 10dB SNR. This validation MSE is saved in a list that stores the mapping between the population member's number  $k$  and the validation MSE associated with the population member's hyperparameters. When all the population members have been used to determine the validation MSE, at 10dB SNR, for the neural network in Table A.1, the algorithm uses the stored validation MSEs to create the next generation of population members. It does this by computing the fitness value associated with each population member. The fitness value is calculated based on the validation MSE of the neural network in Table A.1 and the formula  $\tau[k] = \frac{10^6}{10^{(1000\omega[k]-45)}} \cdot \omega[k]$  is the  $k^{th}$  population member's validation MSE at 10dB SNR.  $\tau[k]$  is the  $k^{th}$  population member's fitness value. The formula for computing the fitness values is found empirically by maximizing the small differences between validation MSE values at 10dB SNR. If we do not non-

linearly amplify the small differences between MSE values, then the selection probabilities will be almost identical for all population members since their fitness values will be very close to each other. For example, if 5 population members A, B, C, D, and F have validation MSE values of 0.04797, 0.0478, 0.048, 0.04632, and 0.04921, respectively, then it is obvious to see that the selection probabilities for population members A, B, C, D, and F will be almost identical if the fitness function does not non-linearly amplify the validation MSE values. However, if we use our empirical formula to calculate the fitness values, we get the following fitness values for population members A, B, C, D, and F: 1071.51, 1584.89, 1000, 47863.01, and 61.66, respectively. The population member F has the worst validation MSE and the lowest fitness value of 61.66. The population member D has the best validation MSE and the highest fitness value of 47863.01. The difference between the validation MSE for population members F and D is only 0.0029. However, the fitness values have a huge difference to avoid equal selection probabilities being calculated in scenarios when the validation MSE values are very close to each other.

After the fitness values are computed, the selection probability per population member is calculated based on each population member's fitness value. The selection probabilities are computed based on the formula  $\rho[k] = \frac{\tau[k]}{\sum_{l=1}^P \tau[l]}$  [32].  $\rho[k]$  is the selection probability of the  $k^{th}$  population member. The selection probability is the likelihood of a population member being randomly selected to mate and produce a child for future generations. Only two parents are randomly selected to produce one child.

When the two parents are selected randomly, a genetic crossover is performed to produce a child. The crossover is done by inheriting the pseudo-random seed value from the first parent and then inheriting the batch size and the learning rate from the second parent. Random genetic mutation may occur with a probability of 0.35 for local mutation and 0.15 for global mutation. Local mutation involves altering a single inherited hyperparameter value of the child. Global mutation involves the altering of all three inherited hyperparameter values of the child. The genetic mutation involves randomly

assigning values to the hyperparameters subject to the constraints stated in Step 1 of Algorithm A.1.

The child produced from the selected parents' mating is then added as a new population member for the next generation. Selecting mating parents and producing children is repeated until the new population size is  $P$ . Only then do we destroy the old population and move on to the next generation of population members.

The process of searching for a global minimum validation MSE is repeated until the number of iterations is equal to the maximum number of generations  $G$ . Only then do we search for the population member  $\Omega_k$ , together with its hyperparameters, that produces the lowest validation MSE evaluated from generation 1 to generation  $G$ . The optimal population member's hyperparameters are then used to fully train the neural network in Table 1 from 16dB to 0dB SNR.

### 3.2 *Training the NN-ML Channel Estimator*

There are two sets of data used in the machine learning algorithm training and testing in Table A.1. The first dataset  $\mathcal{A}$ , is the training dataset, which is made up of 80% of the 10 000 samples generated using the received pilot symbol matrix data from Equation (A.3), and the second dataset  $\mathcal{M}$  is the test dataset, which is the remaining unseen 20% of the 10 000 samples. The datasets are collected for SNR values from 0dB to 16dB. The datasets are generated for both scenarios when  $\alpha = 0$  and  $\alpha = \alpha_{opt}$ . We trained the machine learning algorithm over a wide range of SNR values 0dB to 16dB. The algorithm was trained from 16dB down to 0dB as it was noticed that training it the other way from 0dB to 16dB yielded a poor validation mean squared error per SNR training cycle.

For the training to happen, the received complex pilot symbol matrix from Equation (A.3) is converted into a 2-dimensional data structure, with real numbers, with a single channel. A matrix representation that converts a complex matrix into a 2-dimensional real-valued data structure is used as per [33]. The resulting 2-dimensional data structure is stored in a single element array to create a single channel. Since the received pilot

symbol matrix is  $\mathbb{C}^{N_r \times N}$ , the real-valued matrix is  $\mathbb{R}^{2N_r \times 2N}$  as shown in Equation (A.9) [33]

$$\mathbf{Z}_r = \begin{pmatrix} \Re(\mathbf{Y}_r) & -\Im(\mathbf{Y}_r) \\ \Im(\mathbf{Y}_r) & \Re(\mathbf{Y}_r) \end{pmatrix} \in \mathbb{R}^{2N_r \times 2N} \quad (\text{A.9})$$

where in Equation (A.9):

$$\mathbf{Y}_r = \begin{pmatrix} y_{11} & \cdots & y_{1N} \\ \vdots & \ddots & \vdots \\ y_{N_r 1} & \cdots & y_{N_r N} \end{pmatrix} \in \mathbb{C}^{N_r \times N} .$$

The set of 10 000 samples is made up of the 2-dimensional data  $\mathbf{Z}_r$  with a single channel. To train the machine learning algorithm, using supervised learning, we need the output label data that corresponds to this input training set  $\mathring{\mathbf{A}}$ . The output training data is the actual wireless channel matrix  $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$  as per the method used in [11]. We then convert this wireless channel matrix into a real-valued vector  $\mathbf{h} \in \mathbb{R}^{1 \times 2N_r N_t}$ , which in our case we have  $N_r = 4$  and  $N_t = 2$ , which means  $\mathbf{h} \in \mathbb{R}^{1 \times 16}$ , which makes the output layer of the neural network a 16-neuron layer as per Table A.1. The real-valued entries of the vector  $\mathbf{h}$  are determined by taking the real and imaginary values of the complex entries of the channel matrix  $\mathbf{H}$ .

During training, the set  $\mathring{\mathbf{A}}$  is fed into the machine learning algorithm function  $F(\mathring{\mathbf{A}}, K)$  with hyper-parameters ( $K$ ). This function's output is compared to the output labeled data in vector  $\mathbf{h}$  that corresponds to the actual wireless channel as per [11]. The function  $F(\mathring{\mathbf{A}}, K)$  hyper-parameters ( $K$ ) are tuned using the Adam optimizer with an objective to minimize the validation MSE based on the test data from set  $\mathcal{M}$ . The optimization objective function or loss function is shown in Equation (A.10)

$$\text{MSE Loss} = \frac{1}{\|\mathring{\mathbf{A}}\|} \sum_{\mathbf{x} \in \mathring{\mathbf{A}}} \|F(\mathbf{x}, K) - \mathbf{h}\|_F^2 \quad (\text{A.10})$$

where  $\|\mathring{\mathbf{A}}\|$  is the batch size of the training set. This loss function is used in back-propagation by the Adam optimizer [34] to determine the neural network function

$F(\hat{\mathbf{A}}, K)$  hyper-parameters ( $K$ ), or synaptic weights. During training, the training process looks for the weights that produce the lowest validation MSE, which is determined by evaluating the MSE loss function in Equation (A.10) after each training epoch using the unseen test data set  $\mathcal{M}$ .

### 3.3 Optimal Transmit Power Sharing

Based on Section 2's system model, we derive the optimal power fraction function  $\alpha_{opt} = f(\mathbf{X}_r, N, M, N_t)$ . This subsection is dedicated to deriving the optimal power fraction function and finding the optimal number of pilot symbols to be transmitted for optimal MSE and BER performance.

Inspired by a generalized differential scheme for spatial modulation systems [29], the following derivation is performed to determine the optimal power fraction that minimizes the NN-ML wireless channel estimator's MSE and BER performance. Based on Equations (A.3) and (A.4), we can see that the channel matrix  $\mathbf{H}$  is the common link between Equations (A.3) and (A.4). Manipulating Equation (A.3), we get the following Equation (A.11):

$(\mathbf{Y}_r - \mathbf{N}_r) = \mathbf{H}\mathbf{X}_r$ , further manipulation yields

$(\mathbf{Y}_r - \mathbf{N}_r)\mathbf{X}_r^H = \mathbf{H}\mathbf{X}_r\mathbf{X}_r^H$ , where  $\mathbf{X}_r\mathbf{X}_r^H$  is an invertible square matrix, thus

$$\mathbf{H} = (\mathbf{Y}_r - \mathbf{N}_r)\mathbf{X}_r^H(\mathbf{X}_r\mathbf{X}_r^H)^{-1} \quad (\text{A.11})$$

Substitute Equation (A.11) into Equation (A.4) we get the following:

$\mathbf{Y}_i = ((\mathbf{Y}_r - \mathbf{N}_r)\mathbf{X}_r^H(\mathbf{X}_r\mathbf{X}_r^H)^{-1})\mathbf{X}_i + \mathbf{N}_i$ , and simplifying yields Equation (A.12)

$$\mathbf{Y}_i = \mathbf{Y}_r\mathbf{X}_r^H(\mathbf{X}_r\mathbf{X}_r^H)^{-1}\mathbf{X}_i - \mathbf{N}_r\mathbf{X}_r^H(\mathbf{X}_r\mathbf{X}_r^H)^{-1}\mathbf{X}_i + \mathbf{N}_i \quad (\text{A.12})$$

We can see from Equation (A.12) that the first coefficient of  $\mathbf{X}_i$  is  $\mathbf{Y}_r\mathbf{X}_r^H(\mathbf{X}_r\mathbf{X}_r^H)^{-1}$  and is identical to Equation (A.7), which is the least-squares wireless channel estimate. We can then replace the first coefficient of  $\mathbf{X}_i$  with the generic placeholder for the wireless

channel estimate, which we will call  $\hat{\mathbf{H}}_{est}$ . This changes Equation (A.12) to be represented mathematically as  $\mathbf{Y}_i = \hat{\mathbf{H}}_{est}\mathbf{X}_i - \mathbf{N}_r\mathbf{X}_r^H(\mathbf{X}_r\mathbf{X}_r^H)^{-1}\mathbf{X}_i + \mathbf{N}_i$ .

To find the optimal power fraction that minimizes the channel estimate MSE and BER, we need to derive the equivalent noise power based on a similar method used to derive the equivalent noise power for a generalized differential scheme for spatial modulation systems [29]. From Equation (A.12) it is clear that the equivalent noise power is actually dependent on the noise term  $-\mathbf{N}_r\mathbf{X}_r^H(\mathbf{X}_r\mathbf{X}_r^H)^{-1}\mathbf{X}_i + \mathbf{N}_i$ , for the MIMO pilot assisted wireless channel estimation methods. If  $\hat{\mathbf{H}}_{est}$  is the estimated wireless channel at an instant, then the average signal mean squared error expression,  $argmin_{\hat{\mathbf{H}}_{est}} E\left(\|\mathbf{Y}_i - \hat{\mathbf{H}}_{est}\mathbf{X}_i\|_F^2\right)$ , needs to be minimized for a good channel estimation accuracy where the operator  $E(\cdot)$  is the statistical expectation given that  $\mathbf{X}_i$  is known at the receiver. In order to practically evaluate the accuracy or MSE of the wireless channel estimate, we need to transmit a fixed, known  $\mathbf{X}_i$  information symbol matrix based on the USTLD method and observe the received  $\mathbf{Y}_i$  symbol matrix and evaluate  $E\left(\|\mathbf{Y}_i - \hat{\mathbf{H}}_{est}\mathbf{X}_i\|_F^2\right)$  after having have estimated the wireless channel and obtained  $\hat{\mathbf{H}}_{est}$ . Expanding the minimization of the term  $E\left(\|\mathbf{Y}_i - \hat{\mathbf{H}}_{est}\mathbf{X}_i\|_F^2\right)$ , we get the following mathematical expression:  $argmin_{\hat{\mathbf{H}}_{est}} E\left(\|\mathbf{Y}_i - \hat{\mathbf{H}}_{est}\mathbf{X}_i\|_F^2\right) = argmin_{\hat{\mathbf{H}}_{est}} E\left(\|(\mathbf{H} - \hat{\mathbf{H}}_{est})\mathbf{X}_i + \mathbf{N}_i\|_F^2\right)$ . Using the Frobenius norm property in Equation (A.13.1) and the Cauchy–Bunyakovsky–Schwarz inequality in Equation (A.13.2) [35]

$$\|\mathbf{AB}\|_F \leq \|\mathbf{A}\|_F \|\mathbf{B}\|_F \quad (\text{A.13.1})$$

$$|\text{tr}(\mathbf{AB}^H)| \leq \|\mathbf{A}\|_F \|\mathbf{B}\|_F \quad (\text{A.13.2})$$

And using the Frobenius norm triangle inequality in [35], we get

$\operatorname{argmin}_{\hat{\mathbf{H}}_{est}} E \left( \left\| (\mathbf{H} - \hat{\mathbf{H}}_{est}) \mathbf{X}_i + \mathbf{N}_i \right\|_F^2 \right) \leq \operatorname{argmin}_{\hat{\mathbf{H}}_{est}} E \left( \left\| \mathbf{H} - \hat{\mathbf{H}}_{est} \right\|_F^2 \left\| \mathbf{X}_i \right\|_F^2 + \left\| \mathbf{N}_i \right\|_F^2 + 2 \left\| \mathbf{H} - \hat{\mathbf{H}}_{est} \right\|_F \left\| \mathbf{X}_i \right\|_F \left\| \mathbf{N}_i \right\|_F \right)$ . But  $E(\left\| \mathbf{X}_i \right\|_F) = \lambda$ ,  $E(\left\| \mathbf{N}_i \right\|_F) = \mu$ ,  $E(\left\| \mathbf{X}_i \right\|_F^2) = 2N_t$  and  $E(\left\| \mathbf{N}_i \right\|_F^2) = 2N_r \sigma_i^2$  therefore we can simplify the expression to form Equation (A.14)

$$\operatorname{argmin}_{\hat{\mathbf{H}}_{est}} E \left( \left\| (\mathbf{H} - \hat{\mathbf{H}}_{est}) \mathbf{X}_i + \mathbf{N}_i \right\|_F^2 \right) \leq \operatorname{argmin}_{\hat{\mathbf{H}}_{est}} \left( 2E \left( \left\| \mathbf{H} - \hat{\mathbf{H}}_{est} \right\|_F^2 \right) N_t + 2N_r \sigma_i^2 + 2E \left( \left\| \mathbf{H} - \hat{\mathbf{H}}_{est} \right\|_F \right) \lambda \mu \right) \quad (\text{A.14})$$

where  $N_t$  is the number of transmit antennas and  $N_r$  is the number of receive antennas in the wireless MIMO configuration. We can see from Equation (A.14) that the minimization of  $E \left( \left\| \mathbf{Y}_i - \hat{\mathbf{H}}_{est} \mathbf{X}_i \right\|_F^2 \right)$  implies minimizing the channel estimate MSE  $E \left( \left\| \mathbf{H} - \hat{\mathbf{H}}_{est} \right\|_F^2 \right)$  term. Using this fact, we can then link the channel estimation MSE minimization to the minimization of the equivalent noise power term. Based on Equation (A.12),  $E \left( \left\| \mathbf{Y}_i - \hat{\mathbf{H}}_{est} \mathbf{X}_i \right\|_F^2 \right) = E \left( \left\| -\mathbf{N}_r \mathbf{X}_r^H (\mathbf{X}_r \mathbf{X}_r^H)^{-1} \mathbf{X}_i + \mathbf{N}_i \right\|_F^2 \right)$ . Therefore, we can minimize the average signal mean squared error  $E \left( \left\| \mathbf{Y}_i - \hat{\mathbf{H}}_{est} \mathbf{X}_i \right\|_F^2 \right)$  expression by minimizing its equivalent expression  $E \left( \left\| -\mathbf{N}_r \mathbf{X}_r^H (\mathbf{X}_r \mathbf{X}_r^H)^{-1} \mathbf{X}_i + \mathbf{N}_i \right\|_F^2 \right)$ . But

$$E \left( \left\| -\mathbf{N}_r \mathbf{X}_r^H (\mathbf{X}_r \mathbf{X}_r^H)^{-1} \mathbf{X}_i + \mathbf{N}_i \right\|_F^2 \right) \geq E \left( \left\| \mathbf{N}_i \right\|_F^2 + \left\| \mathbf{N}_r \mathbf{X}_r^H (\mathbf{X}_r \mathbf{X}_r^H)^{-1} \mathbf{X}_i \right\|_F^2 + 2 \left\| \mathbf{N}_i \right\|_F \left\| \mathbf{N}_r \mathbf{X}_r^H (\mathbf{X}_r \mathbf{X}_r^H)^{-1} \mathbf{X}_i \right\|_F \right) \quad (\text{A.15})$$

But we know that  $E \left( \left\| -\mathbf{N}_r \mathbf{X}_r^H (\mathbf{X}_r \mathbf{X}_r^H)^{-1} \mathbf{X}_i + \mathbf{N}_i \right\|_F^2 \right) \geq 0 \Rightarrow E \left( \left\| \mathbf{N}_i \right\|_F^2 \right) + E \left( \left\| \mathbf{N}_r \mathbf{X}_r^H (\mathbf{X}_r \mathbf{X}_r^H)^{-1} \mathbf{X}_i \right\|_F^2 \right) \geq E \left( 2 \left\| \mathbf{N}_i \right\|_F \left\| \mathbf{N}_r \mathbf{X}_r^H (\mathbf{X}_r \mathbf{X}_r^H)^{-1} \mathbf{X}_i \right\|_F \right) \forall \alpha$ .

It is easy to see that when the expression  $E \left( \left\| \mathbf{N}_i \right\|_F^2 \right) + E \left( \left\| \mathbf{N}_r \mathbf{X}_r^H (\mathbf{X}_r \mathbf{X}_r^H)^{-1} \mathbf{X}_i \right\|_F^2 \right)$  increases in value, the expression  $E \left( 2 \left\| \mathbf{N}_i \right\|_F \left\| \mathbf{N}_r \mathbf{X}_r^H (\mathbf{X}_r \mathbf{X}_r^H)^{-1} \mathbf{X}_i \right\|_F \right)$  also increases in value, and the converse is true, which implies that the two expressions are in phase. This means that the mathematical expression,  $E \left( 2 \left\| \mathbf{N}_i \right\|_F \left\| \mathbf{N}_r \mathbf{X}_r^H (\mathbf{X}_r \mathbf{X}_r^H)^{-1} \mathbf{X}_i \right\|_F \right)$ , only shifts or translates the graph of  $E \left( \left\| \mathbf{N}_i \right\|_F^2 \right) + E \left( \left\| \mathbf{N}_r \mathbf{X}_r^H (\mathbf{X}_r \mathbf{X}_r^H)^{-1} \mathbf{X}_i \right\|_F^2 \right)$  vertically downwards

on the Cartesian plane. It does not affect the  $\alpha$  value where the minimum or stationary point occurs. Thus, to minimize the term  $E(\|-\mathbf{N}_r \mathbf{X}_r^H (\mathbf{X}_r \mathbf{X}_r^H)^{-1} \mathbf{X}_i + \mathbf{N}_i\|_F^2)$ , we need to only concentrate on minimizing the term  $E(\|\mathbf{N}_i\|_F^2) + E(\|\mathbf{N}_r \mathbf{X}_r^H (\mathbf{X}_r \mathbf{X}_r^H)^{-1} \mathbf{X}_i\|_F^2)$ .

However, knowing that the total equivalent noise power, henceforth the equivalent noise power, is the total noise power contribution from the noise matrix expressions  $\mathbf{N}_r \mathbf{X}_r^H (\mathbf{X}_r \mathbf{X}_r^H)^{-1} \mathbf{X}_i$  and  $\mathbf{N}_i$ . We can define the equivalent noise power as follows in Equation (A.16):

$$\sigma_{equ}^2 \triangleq E(\|\mathbf{N}_i\|_F^2) + E(\|\mathbf{N}_r \mathbf{X}_r^H (\mathbf{X}_r \mathbf{X}_r^H)^{-1} \mathbf{X}_i\|_F^2) \quad (\text{A.16})$$

As can be seen, that Equation (A.15) contains the equivalent noise power; thus, for us to minimize the channel estimation MSE, we can find the minimum or lowest equivalent noise power. So mathematically, this is noted as follows in Equation (A.17):

$$\underset{\alpha}{\operatorname{argmin}} \sigma_{equ}^2 \Rightarrow \underset{\hat{\mathbf{H}}_{est}}{\operatorname{argmin}} E(\|\mathbf{Y}_i - \hat{\mathbf{H}}_{est} \mathbf{X}_i\|_F^2) \Rightarrow \underset{\hat{\mathbf{H}}_{est}}{\operatorname{argmin}} E(\|\mathbf{H} - \hat{\mathbf{H}}_{est}\|_F^2) \quad (\text{A.17})$$

Based on Equation (A.16), we expand the expression  $E(\|\mathbf{N}_r \mathbf{X}_r^H (\mathbf{X}_r \mathbf{X}_r^H)^{-1} \mathbf{X}_i\|_F^2)$  using the following inequality  $E(\|\mathbf{N}_r \mathbf{X}_r^H (\mathbf{X}_r \mathbf{X}_r^H)^{-1} \mathbf{X}_i\|_F^2) \leq$

$E(\|\mathbf{N}_r\|_F^2 \|\mathbf{X}_r^H (\mathbf{X}_r \mathbf{X}_r^H)^{-1}\|_F^2 \|\mathbf{X}_i\|_F^2)$ . Now let  $\boldsymbol{\theta} = \mathbf{X}_r^H (\mathbf{X}_r \mathbf{X}_r^H)^{-1}$  where  $\boldsymbol{\theta} \in \mathbb{C}^{N \times N_t}$

matrix. The matrix  $\boldsymbol{\theta}$  has constant complex entries containing the fixed transmitted pilot symbol matrix  $\mathbf{X}_r$ . Therefore, the equivalent noise power becomes Equation

(A.18)

$$\sigma_{equ}^2 \leq 2N_r \sigma_i^2 + E(\|\mathbf{N}_r\|_F^2) E(\|\boldsymbol{\theta}\|_F^2) E(\|\mathbf{X}_i\|_F^2) \quad (\text{A.18})$$

since matrices  $\mathbf{N}_r$ ,  $\boldsymbol{\theta}$  and  $\mathbf{X}_i$  are independent. But  $E(\|\mathbf{N}_r\|_F^2) = N_r N \sigma_r^2$ ,  $E(\|\boldsymbol{\theta}\|_F^2) = \|\boldsymbol{\theta}\|_F^2$ , and  $E(\|\mathbf{X}_i\|_F^2) = 2N_t$  thus the equivalent noise power upper bound is given in Equation (A.19)

$$\sigma_{equ}^2 \leq 2N_r\sigma_i^2 + 2N_rN_tN\sigma_r^2\|\boldsymbol{\theta}\|_F^2 \quad (\text{A.19})$$

Using Equations (A.6.1) and (A.6.2), we can get the complete equivalent noise power upper bound as stated in Equation (A.20)

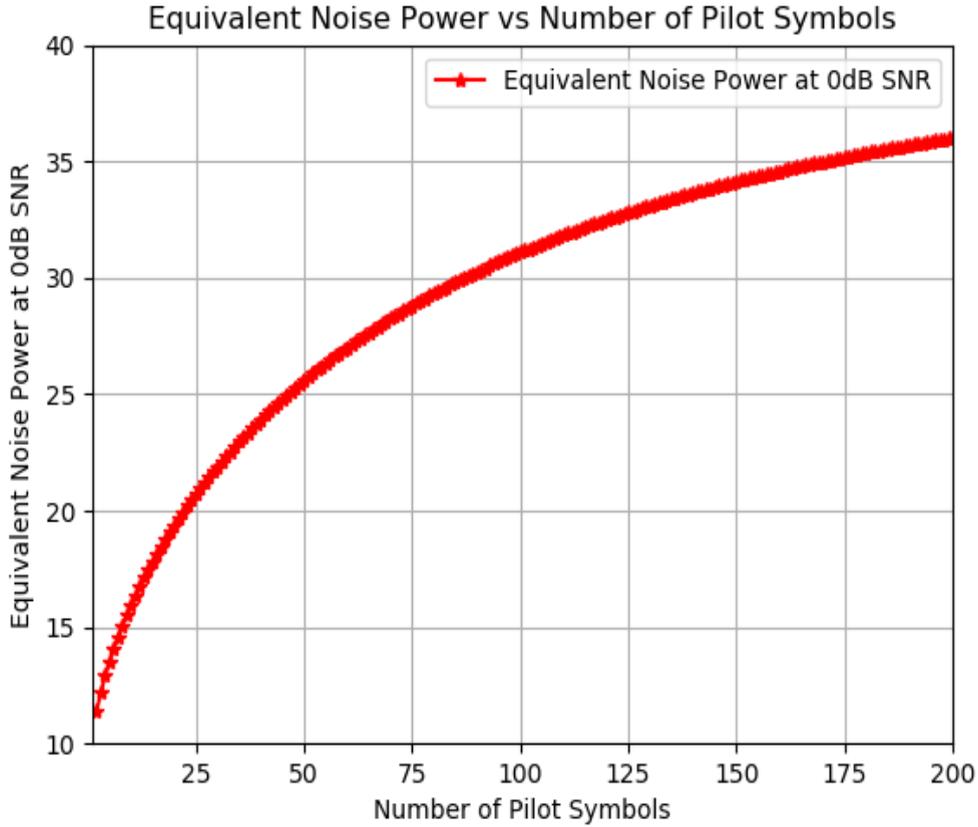
$$\sigma_{equ}^2 \leq \frac{4N_r}{(1-\alpha)\gamma} + \frac{4N_rN_tN\|\boldsymbol{\theta}\|_F^2}{\left(1+\frac{M}{N}\alpha\right)\gamma} \quad (\text{A.20})$$

To get the optimal power fraction that minimizes the equivalent noise power, we rely on the calculus of finding the stationary point of the equivalent noise power with respect to the transmit power fraction.  $\frac{d\sigma_{equ}^2}{d\alpha} = 0$ , solving this leads to the following optimal power fraction as shown in Equation (A.21)

$$\alpha_{opt} = \frac{N\sqrt{MN_t}\|\boldsymbol{\theta}\|_{F-N}}{M+N\sqrt{MN_t}\|\boldsymbol{\theta}\|_F} \quad (\text{A.21})$$

where  $\alpha_{opt}$  is the optimal transmit power fraction,  $M$  is the number of information symbols transmitted per transmit antenna,  $N$  is the number of pilot symbols sent per transmit antenna,  $N_t$  is the number of transmit antennas in the MIMO configuration and  $\|\boldsymbol{\theta}\|_F$  is the Frobenius norm of the matrix  $\boldsymbol{\theta} = \mathbf{X}_r^H(\mathbf{X}_r\mathbf{X}_r^H)^{-1}$ .

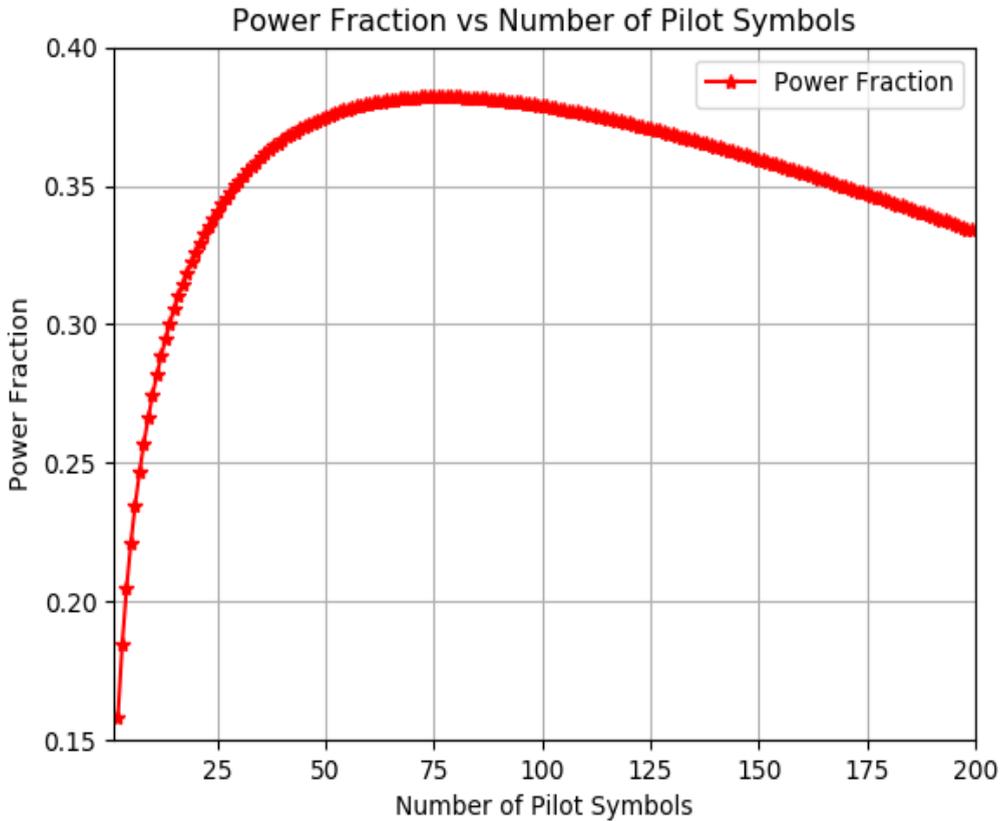
The next objective is to find the optimal number of pilot symbols that must be transmitted over the wireless channel and used for channel estimation. Since it follows that getting a minimal or lowest equivalent noise power translates to minimum channel estimation MSE, we need to select the number of pilot symbols that produce the lowest possible equivalent noise power. The optimal power fraction that minimizes the equivalent noise power is a function of the number of pilot symbols  $N$ . Thus, the critical parameter to select for optimal channel estimation performance is the number of pilot symbols  $N$  since the optimal power fraction can be obtained from Fig. A.3 after finding the optimal  $N$ . We can find the optimal number of pilot symbols  $N$  from Fig. A.2.



**Fig. A. 2 Shows the equivalent noise power at 0dB SNR versus number of pilot symbols  $N$ .**

As can be seen in Fig. A.2, the realistic values for the number of pilot symbols can only be in the range 2 to 200 since the quasi-static channel fading is constant for slightly more than 200 symbols at a time, hence the limit of 200. However, we cannot use  $N = 1$  pilot symbol because the Zadoff-Chu sequence always starts with an element with an amplitude of 1 and a phase of  $0^\circ$ . This means that for  $N = 1$ , we have a singular square matrix created by  $\mathbf{X}_r \mathbf{X}_r^H$ , which is not invertible. We, therefore, can only work with values of  $N$  in the range 2 to 200. This is our search space for our optimal number of pilot symbols. The SNR is set to 0dB because it plays an insignificant role in determining the optimal number of pilot symbols since the SNR is just a scaling factor. Setting the SNR to 0dB is equivalent to setting a scaling factor of 1 on the linear scale.

Fig. A.2 exhibits the fact that the lowest equivalent noise power value is observed at  $N = 2$  pilot symbols. As shown in Section 3.3's derivation, the lowest equivalent noise power value corresponds to the lowest MSE performance. This translates to an optimal power fraction  $\alpha_{opt} \approx 0.16$  based on extrapolation from Fig. A.3 at  $N = 2$ . Fig. A.3 is produced from Equation (A.21) and Fig. A.2 is produced from Equation (A.20) with the transmit power fraction set as  $\alpha = \alpha_{opt}$ .



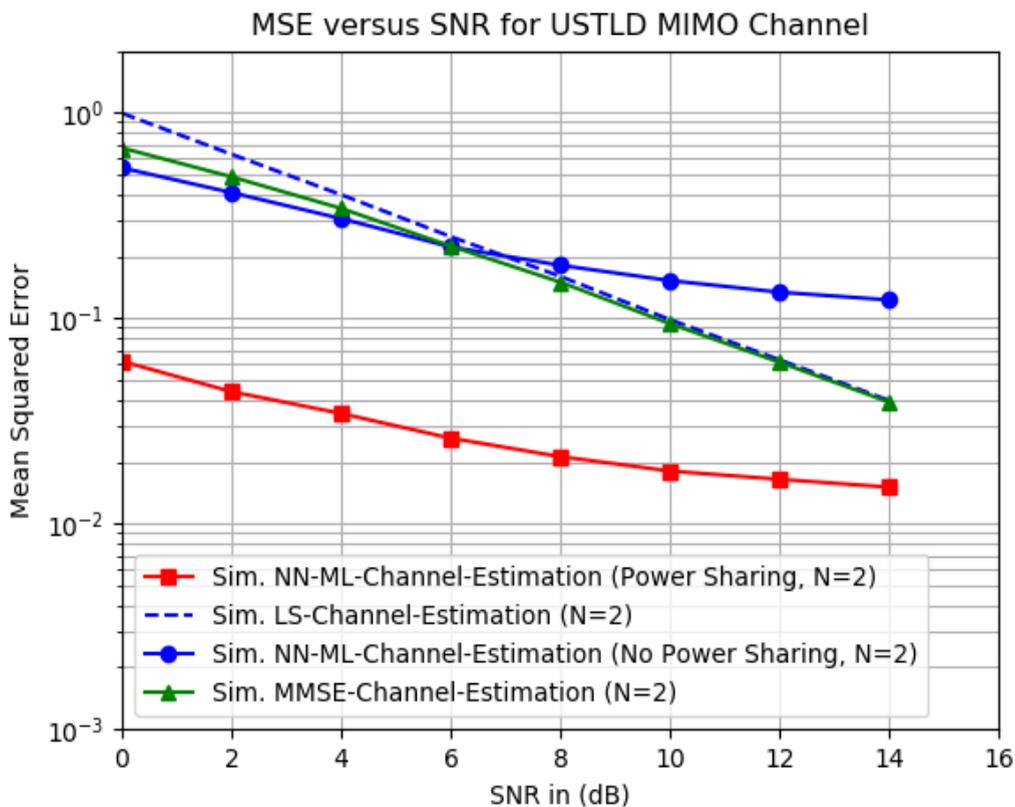
**Fig. A. 3 Shows the power fraction versus number of pilot symbols  $N$ .**

From Fig. A.3, we can see that the optimal power fraction stays within the feasible range of  $[0,1)$  over a different number of pilot symbol values. The other observation is that the power fraction cannot increase and approach unity linearly because the transmit power of the information symbols will become negligible, negatively affecting the BER performance of the transmitted information symbols. Thus whilst increasing the transmit power of the pilot symbols may yield an excellent channel estimation MSE performance,

lowering the transmit power of the information symbols close to 0 will yield inferior BER performances, which will defeat the aim of improving the channel estimation accuracy, thus a perfect balance must be struck to optimize the BER and MSE performance.

## 4 Simulation Results

The Monte-Carlo wireless simulation environment is setup as a  $2 \times 4$  multiple-input-multiple-output (MIMO) wireless channel with Rayleigh quasi-static fading in which the channel gain remains constant for  $200 + N$  symbol durations and changes every  $200 + N$  symbol durations. The wireless transmit and receive antennas are sufficiently spaced enough such that the wireless channels are de-correlated. The number of information symbols transmitted per frame is  $M = 200$  and the number of pilot symbols transmitted per frame is  $N = 2$  or  $N = 10$ . The information and pilot symbols share a fraction of the transmit power based on the optimal power fraction in Equation (A.21) for  $N = 2$ . The pilot symbols are generated using the Zadoff-Chu sequence in Equation (A.1) and based on the number of pilot symbols  $N = 2$  or  $N = 10$ . The information symbol modulation order used in the simulation is 16-QAM and 16-PSK. The average power constraint for the 16-QAM and 16-PSK symbols is set to 1. The NN-ML-based algorithm model architecture seen in Table A.1 is loaded into the simulation environment, and the saved optimal synaptic weights are loaded into the machine learning model. The channel estimation algorithms used are the LS, MMSE, NN-ML without power-sharing, and NN-ML with power-sharing. The Monte-Carlo simulation determines the channel estimation algorithms' MSE and BER performance over 0dB to 14dB SNR.

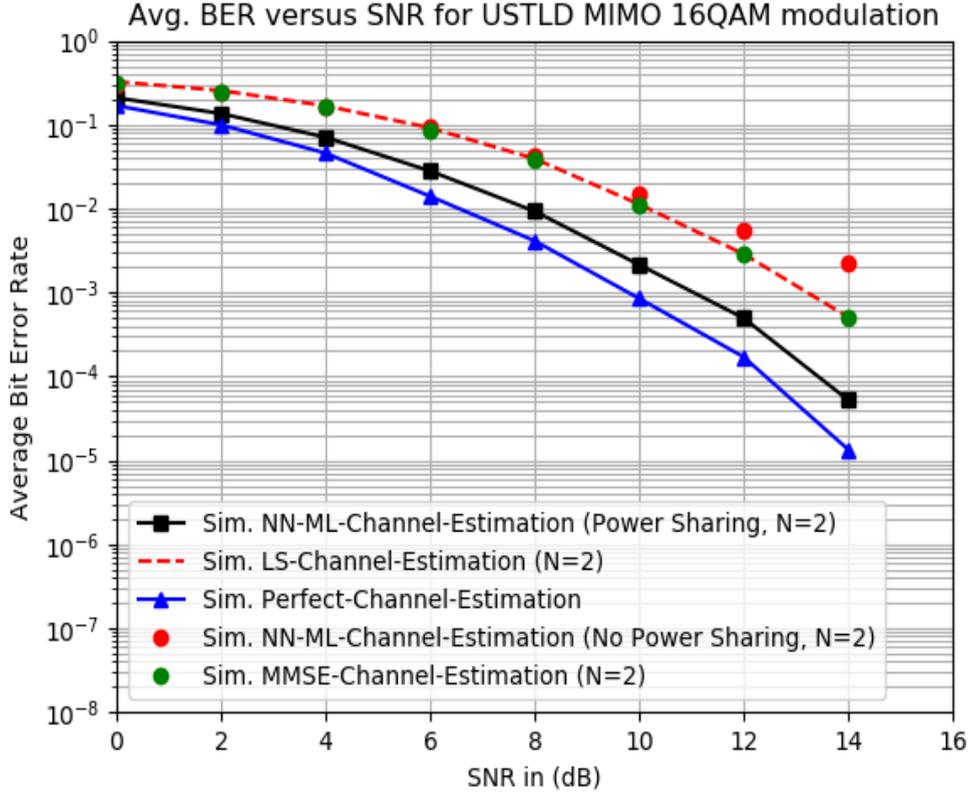


**Fig. A. 4 Shows the MSE performance of the  $2 \times 4$  MIMO NN-ML channel estimation vs traditional channel estimation methods.**

From Fig. A.4 we can see that the NN-ML channel estimator without transmit power-sharing underperforms the traditional channel estimation methods LS and MMSE at high SNR. The reason for this is that the hyperparameters in Table A.1 are searched for in a multivariable landscape with the objective to find the global minimum validation MSE at 10dB SNR. Because the landscape is multivariable, it lends itself to the possibility of having multiple local stationary points and a single global stationary point. The genetic algorithm (GA) used in Algorithm A.1 tries to search for the global stationary point in the multivariable landscape with no guarantees of finding the global stationary point. Our GA runs for only 100 evolutionary generations, with a population size of 10, and this limits the number of permutations of hyperparameter values tested on the neural network. Thus, it restricts the search space for the global stationary point. The

hyperparameters found in Table A.1 for the case  $\alpha = 0$ , are clearly suboptimal based on the poor MSE performance at high SNR. The higher SNR range is sensitive to the channel estimator's systematic errors as opposed to the low SNR range.

We also see from Fig. A.4 that the NN-ML channel estimation method has an improved MSE performance when transmit power-sharing is used to improve its performance. From Fig. A.4, the added transmit power fraction for the pilot symbols improves the MSE performance. It is also interesting to note that for the defined SNR range, the NN-ML channel estimator with transmit power-sharing outperforms the traditional LS and MMSE methods. At  $\text{MSE} = 6 \times 10^{-2}$ , we have an approximately 12dB gain over the traditional LS and MMSE channel estimation when using the NN-ML with power-sharing channel estimation algorithm. This implies that we can save transmit power with the NN-ML with the power-sharing method; we can also do blind channel estimation without knowledge of the transmitted pilot symbols and/or channel autocorrelation statistics or the received noise variance required by the traditional channel estimation methods.



**Fig. A. 5 Shows the BER performance of the  $2 \times 4$  MIMO NN-ML channel estimation vs traditional channel estimation methods at the same number of pilot symbols for 16-QAM modulation.**

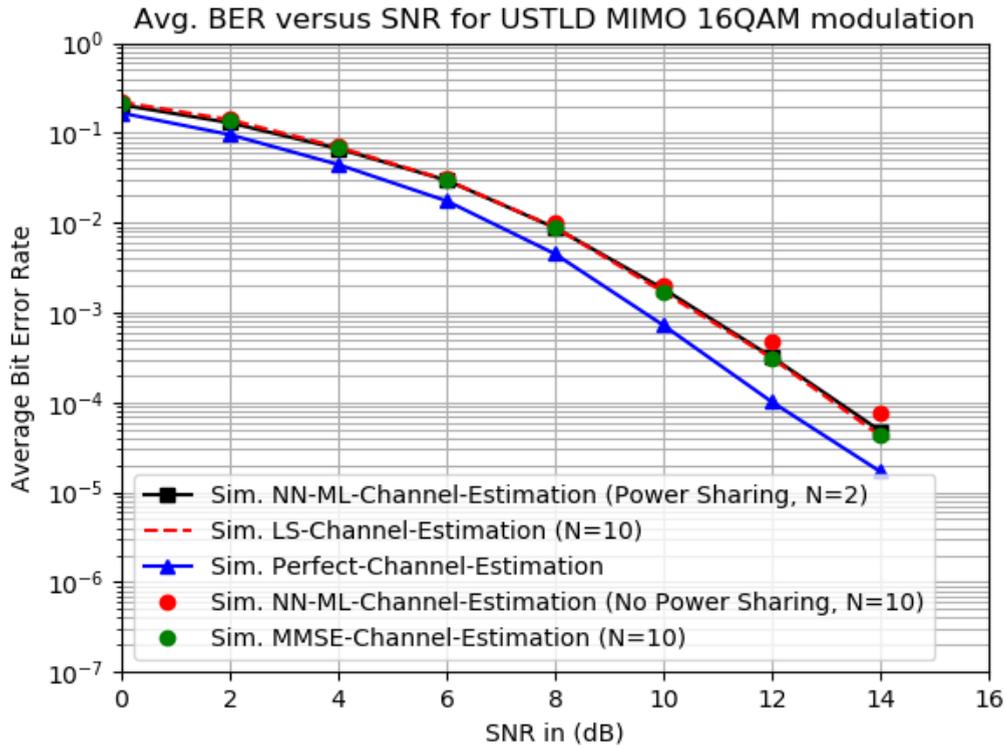
As shown in Fig. A.5, the NN-ML channel estimator's BER performance with transmit power-sharing is the best performing as expected from the MSE accuracy shown in Fig. A.4. This shows that minimizing the equivalent noise power does, in fact, minimize the channel estimation MSE performance as shown in Fig. A.4 and the signal MSE performance as shown in Fig. A.5. The signal MSE performance is linked to the maximum likelihood detector performance. Hence, the NN-ML channel estimator's BER performance with transmit power-sharing has the best BER performance since its equivalent noise power is minimized relative to the other channel estimation algorithms.

There is a loss in diversity at high SNR for the NN-ML channel estimator without transmit power-sharing as the hyperparameters selected for the case  $\alpha = 0$  are

suboptimal. Fig. A.4 MSE performance shows that at high SNR the NN-ML channel estimator without transmit power-sharing has poor performance relative to the other channel estimators. This poor MSE performance has an impact on the BER performance, as shown in Fig. A.5. For the NN-ML channel estimator with transmit power-sharing, there is no loss in diversity because the hyperparameters selected for the case when  $\alpha = \alpha_{opt}$  are near-optimal. The approach of selecting hyperparameters using a GA does not guarantee that the stationary points found are globally optimal.

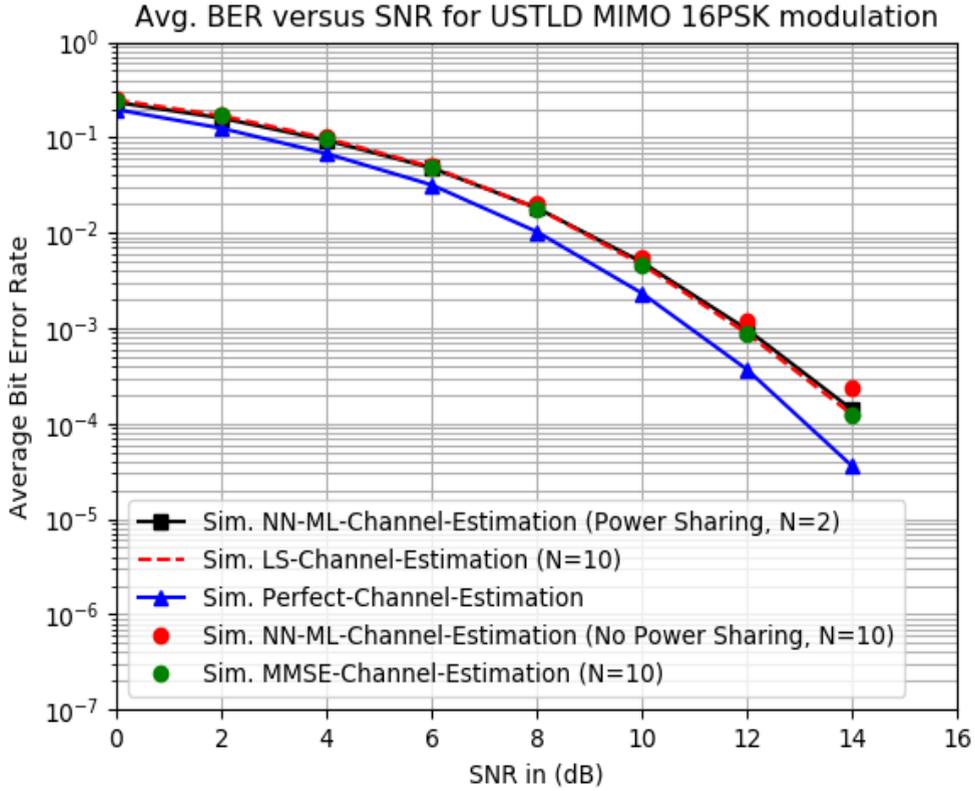
We also observe a near 2dB BER performance gain between the NN-ML channel estimator with transmit power-sharing and the traditional LS and MMSE channel estimation methods. This implies that the NN-ML channel estimator with transmit power-sharing enables a good channel estimation accuracy and link reliability relative to the traditional channel estimation algorithms whilst using a minimal number of pilot symbols to estimate the wireless channel. As shown in Fig. A.6, the NN-ML channel estimator with transmit power-sharing has the same BER performance, at  $N = 2$ , with the traditional LS and MMSE channel estimation methods at  $N = 10$ . This means that we need 8 extra pilot symbols for the LS and MMSE channel estimation methods to deliver a similar BER performance as the NN-ML channel estimator with transmit power-sharing. That is a waste of expensive wireless channel bandwidth, which should be used to transmit information symbols.

The other observation is that the NN-ML channel estimator without transmit power-sharing, at higher  $N$  values, as observed in Fig. A.6, has a BER performance that approaches that of the traditional LS and MMSE methods. This is advantageous because totally blind channel estimation can be achieved by this NN-ML method as it only needs the observed/received pilot symbols matrix to perform channel estimation. This means that it does not need to know the wireless channel second-order statistics, nor does it need to know the noise variance. Neither does it need to know the transmitted pilot symbol matrix, unlike the traditional channel estimation methods that require this prior knowledge.



**Fig. A. 6 Shows the BER performance of the  $2 \times 4$  MIMO NN-ML channel estimation vs traditional channel estimation methods at different number of pilot symbols for 16-QAM modulation.**

In Fig. A.7, we see that the same performance gains achieved in Fig. A.6 for 16-QAM modulation apply to 16-PSK modulation. This implies that 16-QAM or 16-PSK USTLD modulation in conjunction with NN-ML channel estimator with transmit power-sharing can achieve a comparable BER performance to the traditional channel estimation methods but at 20% of the bandwidth required by the traditional channel estimators LS and MMSE.



**Fig. A. 7 Shows the BER performance of the  $2 \times 4$  MIMO NN-ML channel estimation vs traditional channel estimation methods at different number of pilot symbols for 16-PSK modulation.**

## 5 Conclusion

The power-sharing method improves the NN-ML channel estimation MSE accuracy relative to the NN-ML method without transmit power-sharing. The MSE performance of the NN-ML channel estimator algorithm with transmit power-sharing is very good throughout the SNR range relative to the traditional LS and MMSE channel estimation methods. The NN-ML channel estimator with transmit power-sharing MSE performance, at  $\text{MSE} = 6 \times 10^{-2}$ , has an approximately 12dB gain over the traditional LS and MMSE channel estimation methods. The NN-ML algorithm does not require knowledge of the channel autocorrelation statistics and the noise variance to estimate the wireless channel. This implies that the NN-ML algorithm with power-sharing can be used for wireless channel estimation where the transmitted pilot symbols, channel

second-order statistics and receiver noise variance are unknown. Another inference from the results is that, since the optimal number of pilot symbols is only 2, it means we can achieve high channel estimation MSE/BER accuracy whilst saving expensive channel bandwidth since the traditional channel estimation algorithms will need a higher number of pilot symbols to achieve similar MSE/BER performance. From the simulation results, the proposed NN-ML channel estimator with transmit power-sharing requires only 20% of the bandwidth utilized by LS and MMSE to achieve the same BER performance for 16-QAM and 16-PSK USTLD modulation.

## References

- [1]. H. Xu, K. Govindasamy, N. Pillay, “Uncoded Space-Time Labelling Diversity”, IEEE Communications Letters, Volume: 20, Issue: 8, 2016.
- [2]. S. A. Alamouti, “Simple transmit diversity technique for wireless communications”, IEEE J. Sel. Areas Commun., vol. 16, no. 8, pp. 1451–1458, 329, Oct. 1998.
- [3]. S. Patel, T. Quazi, H. Xu. “A Genetic Algorithm for Designing Uncoded Space-Time Labelling Diversity Mapper”, IEEE International Workshop on Signal Processing Systems (SiPS), 2018.
- [4]. D. Ayanda, H. Xu, “A Unified Error Analysis of Uncoded Space-Time Labeling Diversity with Three Transmit Antennas in Rician Fading Channels”, IEEE 2nd Wireless Africa Conference (WAC), 2019.
- [5] N. Pillay and H. Xu, "Uncoded Space-Time Labeling Diversity—Application of Media-Based Modulation with RF Mirrors," IEEE Communications Letters, vol. 22, no. 2, pp. 272-275, Feb. 2018.
- [6] T. Quazi, H. Xu. “SSD-enhanced uncoded space-time labeling diversity”, Int J Commun. Syst. Volume 31, Issue 11, 2018.
- [7] B. Dlodlo, H. Xu. “Trellis code-aided high-rate  $M$ -QAM space-time labeling diversity using a unitary expansion”, Int J Commun Syst. Volume 31, Issue 11, 2018.
- [8] D. Ayanda, H. Xu, N. Pillay. “Uncoded  $M$ -ary quadrature amplitude modulation space-time labeling diversity with three transmit antennas”, Int J Commun Syst. Volume 31, Issue 18, 2018

- [9] D. Ayanda, S. Mughal and K. A. Abdulsalam, "Performance Analysis of Rectangular QAM Uncoded Space-Time Labeling Diversity over Nakagami-m Fading Channels," IEEE Microwave Theory and Techniques in Wireless Communications (MTTW), Riga, Latvia, pp. 163-167, 2020.
- [10] D. Ayanda, H. Xu and S. Mughal, "High-Density M-QAM Mapper Design for Uncoded Space-Time Labeling Diversity with Three Transmit Antennas over Nakagami-m Fading Channels," IEEE AFRICON, Accra, Ghana, pp. 1-7, 2019.
- [11]. M. Soltani, V. Pourahmadi, A. Mirzaei, H. Sheikhzadeh, "Deep Learning-Based Channel Estimation", IEEE Communications Letters, Volume: 23, Issue: 4, 2019.
- [12]. H. He, C. Wen, S. Jin, G. Li, "Deep Learning-based Channel Estimation for BeamSpace mmWave Massive MIMO Systems", IEEE Wireless Communications Letters, Volume: 7, Issue: 5, 2018.
- [13]. Y. Yang, F. Gao, X. Ma, S. Zhang, "Deep Learning-Based Channel Estimation for Doubly Selective Fading Channels", IEEE Access, Volume: 7, 2019.
- [14] S. Gao, P. Dong, Z. Pan and G. Y. Li, "Deep Learning Based Channel Estimation for Massive MIMO With Mixed-Resolution ADCs," IEEE Communications Letters, vol. 23, no. 11, pp. 1989-1993, Nov. 2019.
- [15] J. Kang, C. Chun and I. Kim, "Deep-Learning-Based Channel Estimation for Wireless Energy Transfer," IEEE Communications Letters, vol. 22, no. 11, pp. 2310-2313, Nov. 2018.
- [16] H. He, C. Wen, S. Jin and G. Y. Li, "Deep Learning-Based Channel Estimation for BeamSpace mmWave Massive MIMO Systems," IEEE Wireless Communications Letters, vol. 7, no. 5, pp. 852-855, Oct. 2018.
- [17] Q. Bai, J. Wang, Y. Zhang and J. Song, "Deep Learning-Based Channel Estimation Algorithm Over Time Selective Fading Channels," IEEE Transactions on Cognitive Communications and Networking, vol. 6, no. 1, pp. 125-134, March 2020.
- [18] H. Ye, G. Y. Li and B. Juang, "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems," IEEE Wireless Communications Letters, vol. 7, no. 1, pp. 114-117, Feb. 2018.

- [19] X. Cheng, D. Liu, C. Wang, S. Yan and Z. Zhu, "Deep Learning-Based Channel Estimation and Equalization Scheme for FBMC/OQAM Systems," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 881-884, June 2019,
- [20] Y. Liao, Y. Hua, X. Dai, H. Yao and X. Yang, "ChanEstNet: A Deep Learning Based Channel Estimation for High-Speed Scenarios," *ICC 2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, pp. 1-6, 2019.
- [21] J. P. Nair and R. V. Raja Kumar, "A bandwidth efficient channel estimation method using superimposed training for MIMO-OFDM systems," *TENCON 2008 - 2008 IEEE Region 10 Conference*, Hyderabad, pp. 1-5, 2008.
- [22] W. G. Jeon, K. H. Paik and Y. S. Cho, "An efficient channel estimation technique for OFDM systems with transmitter diversity," *11th IEEE International Symposium on Personal Indoor and Mobile Radio Communications. PIMRC 2000. Proceedings (Cat. No.00TH8525)*, London, UK, pp. 1246-1250, 2000.
- [23] C. Shin, R. W. Heath and E. J. Powers, "Blind Channel Estimation for MIMO-OFDM Systems," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 2, pp. 670-685, March 2007.
- [24] K. Josiam and D. Rajan, "Bandwidth Efficient Channel Estimation Using Super-Imposed Pilots in OFDM Systems", *IEEE Transaction on Wireless Communications*, Vol. 6, No. 6, June 2007.
- [25] J. Siew *et al.*, "A bandwidth efficient channel estimation algorithm for MIMO-SCFDE," *2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No.03CH37484)*, Orlando, FL, pp. 1142-1146, 2003.
- [26] A. Koochian, H. Mehrpouyan, M. Ahmadian and M. Azarbad, "Bandwidth efficient channel estimation for full duplex communication systems," *2015 IEEE International Conference on Communications (ICC)*, London, pp. 4710-4714, 2015.
- [27]. M. Pukkila, "Channel Estimation Modeling. Postgraduate Course in Radio-Communications", *Nokia Research Center*, Fall 2000.
- [28]. C. R. Murthy, A. K. Jagannatham, B. D. Rao, "Training-Based and Semi-Blind Channel Estimation for MIMO Systems with Maximum Ratio Transmission", *IEEE Transactions on Signal Processing*, Vol. 54, Issue No. 7, 2006.

- [29]. K. Kadathlal, H. Xu, N. Pillay, “Generalised differential scheme for spatial modulation systems”, IET Communications, Volume: 11, Issue: 13, 2017.
- [30]. M. PremKumar, M. P. Chitra, M. Arun, M. S. Saravanan, “Least Squares based Channel Estimation Approach and Bit Error Rate analysis of Cognitive Radio”, International Conference on Robotics, Automation, Control and Embedded Systems, India, February 2015.
- [31]. M. Simko, C. Mehlhruer, M. Wrulich, M. Rupp. “Doubly Dispersive Channel Estimation with Scalable Complexity”, IEEE International ITG Workshop on Smart Antennas (WSA), 2010.
- [32]. D. E. Goldberg, “Genetic Algorithms in Search, Optimization and Machine Learning”, Addison-Wesley, 1989.
- [33]. N. Samuel, T. Diskin, A. Wiesel, “Learning to detect”, IEEE Transactions on Signal Processing, Volume: 67, Issue: 10, May15 2019.
- [34]. D. Kingma, J. Ba, “Adam: A method for stochastic optimization”, December 2014.
- [35]. J. M. Steele, “The Cauchy–Schwarz Master Class: an Introduction to the Art of Mathematical Inequalities”, The Mathematical Association of America, 2004.

## **Paper B**

### **Low Complexity Golden Code Analytical and Deep Learning-Based Sphere-Decoders for High-Density M-QAM**

Bhekisizwe Mthethwa and Hongjun Xu

IEEE Access, vol. 10, pp. 6940-6953,  
January 2022.

## Abstract

In this paper, we develop low complexity Golden code sphere-decoding (SD) algorithms for high-density M-ary quadrature amplitude modulation (M-QAM). We define the high-density M-QAM as having modulation orders ( $M$ ) of at least 64, i.e.  $M \geq 64$ . High-density M-QAM symbols deliver high data rates under good wireless channels. Future wireless systems must deliver high data rates and simultaneously low end-to-end latency. However, higher M-QAM modulation orders increase the Golden code SD search breadth, thus increasing decoding latency. We, therefore, propose two forms of low complexity Golden code SD to achieve low decoding latency while maintaining the near-optimal SD bit-error rate (BER). The proposed low complexity SD algorithms are based on the SD with sorted detection subsets (SD-SDS). The literature shows the SD-SDS to achieve lower detection complexity relative to the Schnorr-Euchner SD (SE-SD). The first form of the proposed Golden code SD is the SD-SDS-Descend algorithm with instantaneously varying subset lengths and a search tree search order sorted based on the worst-first search strategy. The second form of the proposed Golden code SD is an SD-SDS algorithm called SD-SDS-ES-DNN with a deep learning-based early stopping search criterion. Our proposed algorithms achieve at most 57% and 70% reduction in Golden code decoding latency relative to SD-SDS, at low SNR, for 64-QAM and 256-QAM, respectively. At high SNR, the proposed algorithms achieve 40% and 37% in Golden code decoding latency reduction relative to the SD-SDS for 64-QAM and 256-QAM, respectively. The decoding latency reduction is achieved while maintaining near-optimal BER performances.

# 1 Introduction

With the high demand for communication services that require high data throughputs and low end-to-end latency, coupled with the sharp increase of mobile devices depending on wireless communications, the wireless communication literature proposes various multiple-input multiple-output (MIMO) architectures to cater to these demands. Wireless MIMO technology offers high data throughputs and link reliability through spatial multiplexing and spatial diversity, respectively [1]. The spatial diversity offers the benefit of creating high-reliability wireless links. However, the wireless link reliability is enhanced using space-time block coding (STBC) signal processing at the transmitter, which adds time diversity to MIMO spatial diversity. These STBC schemes create full-diversity wireless links by transmitting replicas of the data symbols over two or more timeslots. One such scheme is the non-orthogonal STBC Golden code [2].

Golden code is a full-rate full-diversity STBC scheme relevant to meet the demands for high data throughput and link reliability. Golden code offers not only an added time diversity over the space diversity of MIMO, but it also adds spatial multiplexing gain over and above the spatial multiplexing inherent in wireless MIMO. Golden code further improves the data throughputs offered by wireless MIMO over and above enhancing the link reliability. It, however, has the drawback, which other non-orthogonal STBC schemes have, of having nonlinear maximum likelihood (ML) detection at the receiver side [3]. In the case of the Golden code, the optimal ML detector has a detection complexity of  $O(M^4)$  [4]. This nonlinear detection complexity of order  $O(M^4)$ , where  $M$  is the M-QAM modulation order, has the negative effect of high decoding latency, increasing end-to-end latency. Golden code has low adoption in modern wireless standards relative to the orthogonal STBC scheme called Alamouti [5]. Another non-orthogonal STBC scheme is the half-rate full-diversity uncoded space-time labeling diversity (USTLD) with nonlinear optimal ML detection [6].

The orthogonal STBC schemes exploit their orthogonality property to deliver optimal linear ML,  $O(M)$ , detection complexity at the receiver side in block fading channels [3]. For example, the Alamouti is a half-rate full-diversity orthogonal STBC scheme, yet it is implemented in the various WiFi [7] and LTE [8] wireless standards. Golden code is incorporated in the now-

defunct WiMAX standard [9]. Despite the Golden code having spatial multiplexing gain over the Alamouti scheme, its popularity in wireless standards is low, possibly due to the higher detection complexity than the Alamouti linear ML detector.

The literature proposes various novel lower detection complexity algorithms to lower the Golden code detection complexity, hence decoding latency. In [4], the authors propose a fast ML detection of Golden code using a sphere-decoder (SD) with a search tree with reduced dimensions. In [10], the authors manage to reduce the Golden code detection complexity to  $O(M^{1.5})$  at the expense of losing 1 dB signal-to-noise ratio (SNR) relative to the optimal ML detector. The authors introduced fast ML detection in [11], and their detection scheme achieves a detection complexity order of  $O(M^2)$  with near-optimal bit-error-rate (BER) performance. In [12], the authors propose the low complexity Schnorr-Euchner SD (SE-SD) as an SD variant for the Golden code detection. The SE-SD algorithm does not require the re-adjusting or increasing of the search radius compared to the traditional SD [13], which increases its search radius when there are no lattice points found inside the hypersphere. From the literature, it is known that SD detection complexity depends on the search tree's search breadth and depth [14]. The authors in [15] reduce the search breadth of the Golden code SD by creating detection subsets (SD-DS) of the full M-QAM signal cardinality. The SD-DS detection strategy is shown to achieve lower detection complexity relative to the SE-SD algorithm while exhibiting near-optimal BER performances. The SD with sorted detection subsets (SD-SDS) is ventilated in [16], where the M-QAM signal constellation candidate symbols are sorted in ascending order based on which symbols are closest to the estimated symbols detected by the sub-optimal QR decoder. The furthest symbols from the estimated M-QAM symbols are the least likely transmitted symbols. Hence the SD-SDS algorithm creates the detection subsets by rejecting the candidate symbols furthest away from the estimated M-QAM symbols. This strategy reduces the signal cardinality, hence SD search breadth of SD-SDS, and achieves a detection complexity that is 1 order lower than the SD-DS algorithm in [15].

Recently, deep learning has been applied to lower the detection complexity of the SD algorithm for large MIMO architectures. A deep learning algorithm is introduced in [17] that predicts the number of lattice points inside the SD hypersphere. The prediction is based on the SD initial

radius, which is reduced until the number of predicted lattice points inside the hypersphere is small. With this sufficiently small initial radius, the SD algorithm is initiated, and hence lower detection complexity is achieved since the SD complexity also depends on the value of the initial radius. In [18], the authors propose a deep learning-based initial radius predictor algorithm that uses the instantaneous wireless channel conditions and noise statistics to predict the initial radius for SD in large MIMO. The traditional SD algorithm calculates the initial radius based on average channel conditions. The calculation of the initial radius based on the average channel conditions has a disadvantage. When the instantaneous channel is good, the SD will have many lattice points inside the hypersphere due to the fixed initial radius, which depends on average channel conditions. The authors in [19] propose a low complexity deep learning-based SD for large MIMO. This deep learning-based SD algorithm provides low complexity offline training and online decoding compared to the deep learning-based SD algorithms in the literature. In [20], the authors propose a deep learning-based SD minimum path metric predictor for the subtrees. These minimum path metrics are used for early search termination for candidates on the SD search tree. The algorithm is developed for a large MIMO architecture and achieves considerable low detection complexity while achieving near-optimal BER performances.

### ***1.1 Motivation***

The low complexity SD-SDS in [16] is shown to achieve lower detection complexity relative to the SD-DS in [15] and the SE-SD variant described in [12]. However, the SD-SDS detection subset lengths are set based on the average SNR values, as shown in [16, Table 2]. Therefore, the SD-SDS fixed-length detection subsets leave room for further reduction in subset lengths based on the instantaneous channel and noise statistics. Good instantaneous channel and noise statistics may prompt even shorter subset lengths relative to the fixed lengths, as we do not need to search through as many symbol candidates under such conditions. Therefore, we are motivated to propose instantaneously varying subset lengths that vary based on the instantaneous channel quality. The subset lengths can be shortened relative to the average SNR-based subset lengths determined in [16, Table 2] at high instantaneous SNR. Shorter subset lengths shorten the SD-SDS search breadth and thus reduce decoding latency. The SD-SDS algorithm also exhibits another opportunity for reduction in detection complexity by ordering the search order of the SD-SDS search tree. The SD-SDS in [16] currently has a search order that is not ordered based on any instantaneous channel quality. As shown in the paper, a fixed

search order has a disadvantage at low SNR since the SD-SDS algorithm has a detection complexity dominated by the detection complexity at search layer 1 of the search tree. It thus motivates us to propose a worst-first search strategy that ensures that the candidate symbol subset with the most petite subset length is always used at search layer 1. This lowers the decoding latency at low SNR, as shown in the paper.

The SE-SD variant described in [12] orders the search tree search order or wireless channel matrix columns in ascending order, based on the instantaneous wireless fading power. In the search order, the M-QAM symbol that experiences the highest instantaneous wireless fading power is detected first in the SD search tree. However, the metric used to sort the search tree search order is based on the instantaneous wireless fading power and is not an accurate metric for determining instantaneous channel quality. The instantaneous noise statistics are not factored in by [12]. The approach in [12] is valid for high SNR as the wireless fading power dominates the performance at high SNR since the average noise power is very low. However, noise statistics dominate the system performance at low SNR as the average noise power is very large. Therefore, we are motivated to propose a metric that factors in the instantaneous noise power and wireless fading power to sort the SD-SDS search tree search order.

Deep learning-based low complexity SD algorithms are proposed in [17-20]. However, these deep learning algorithms are specifically designed to lower the detection complexity of large MIMO SD systems where  $N_t = N_r \geq 8$ . The number of transmit and receive antennas in a MIMO configuration are defined as  $N_t$  and  $N_r$ , respectively. The deep learning-based SD algorithms also reduce complexity in the traditional SD which is a high complexity decoding algorithm. The proposed deep neural networks (DNN) in [19-20] are very complex for small MIMO environments, such as  $N_t = 2$ . Further, in [20], the authors rely on the large MIMO property of channel hardening to design the DNN architecture that predicts the minimum path metrics for the sub-trees. These predicted minimum path metrics are used to initiate early termination of the SD search. However, this solution will not apply to small MIMO channels as the assumption of channel hardening does not hold. The other drawback of [20] is the DNN architecture complexity. The hidden layer is set to have  $2N_t + 2M$  neurons, and the output layer has  $M$  neurons. It is easy to see that for the high-density M-QAM,  $M \geq 64$  contexts, the DNN

complexity will increase the decoding latency for the small MIMO low complexity SD-SDS-based decoders. Therefore, we are motivated to propose an SD-SDS search tree early stopping deep learning-based algorithm with a low inference time DNN architecture that is invariant to the M-QAM modulation order. This DNN algorithm prematurely terminates the SD-SDS search under learned channel conditions. This has the advantage of lowering the decoding latency of the Golden code SD-SDS search tree.

## 1.2 Contributions

This paper proposes two forms of low complexity Golden code SD-based algorithms. We present analytical algorithms that further reduce the decoding latency of the low complexity SD-SDS. We propose a deep learning-based early stopping algorithm that prematurely terminates the SD-SDS search under specific instantaneous channel conditions. Based on the literature survey, none of the research has attempted to reduce the decoding latency of the low complexity Golden code SD-SDS algorithm in a small MIMO environment i.e  $N_t = 2$  and  $N_r \in [N_t: 8]$ . The reduction in decoding latency is necessary for high-density M-QAM modulation as future wireless standards will require the use of high-density M-QAM for faster data rates but at the same time with a low end-to-end latency constraint. High-density M-QAM increases the search breadth of the Golden code SD-SDS search tree, increasing decoding latency and negatively affecting the end-to-end latency. Our main contributions of the paper are listed as follows:

- We propose a simple metric that more accurately describes the channel quality compared to the instantaneous wireless fading power described in [12]. Not only does our proposed metric consider the wireless fading power gain, but it also indirectly considers the instantaneous noise power.
- We propose a heuristic approach to instantaneously set the sorted candidate symbol subset lengths based on the proposed simple metric used to measure the instantaneous channel quality. The sorted candidate symbol subset lengths are not necessarily identical for each of the estimated M-QAM symbols,  $\hat{x}_q \forall q \in [1: 4]$ , as they experience different wireless fading power and noise power. The instantaneously varying subset lengths are shorter than the average SNR-based fixed subset lengths at high instantaneous SNR. This lowers the decoding latency of the SD-SDS search tree.
- We exploit to our advantage the instantaneously varying wireless channel quality for each estimated M-QAM symbol to sort the search tree search order either in ascending or

descending order based on the proposed channel quality metric. In literature, the search tree search order sorting has mainly been in ascending order or best-first strategy [12]. We show in this paper that the worst-first strategy is beneficial in the low SNR regions and further reduces decoding latency relative to the best-first strategy.

- We finally propose a novel early stopping deep learning-based SD-SDS algorithm that takes advantage of the sorted candidate symbols in the subsets. The candidate symbols are sorted from the most likely transmitted symbol to the least likely transmitted symbol for each search layer in the search tree. We thus take advantage of this and the depth-first search strategy to prematurely terminate the search on the first lattice point found inside the hypersphere. This termination only happens when the instantaneous channel conditions are good. The DNN developed in this paper detects when these channel conditions are good enough to perform early stopping. In literature, an early termination DNN algorithm is developed, for large MIMO [20], using the property of channel hardening. This property does not apply in our context of small MIMO. Over and above that, the DNN architecture in [20] is too complex for our small MIMO low complexity SD-SDS high-density M-QAM environment.

The remainder of this paper is organized as follows: in Section 2, the system model of the paper is presented. In Section 3, we present the theoretical overview of the SD-SDS algorithm. In Section 4, we present the proposed low complexity analytical SD based algorithm. Section 5 presents the low complexity Deep Learning-based SD algorithm. Section 6 presents the Simulations results and discussion. Section 7 concludes the paper.

*Notation:* Bold lowercase letters are used for vectors and bold uppercase for matrices.  $(\cdot)^T$   $(\cdot)^H$ ,  $|\cdot|$ ,  $\|\cdot\|$  and  $\|\cdot\|_F$  represent the Transpose, Hermitian, Absolute Value, Euclidean norm and Frobenius norm operations, respectively. The functions  $\Re(\cdot)$  and  $\Im(\cdot)$  are the real and imaginary components of a complex number, respectively.  $j$  is a complex number. The statistical average is represented by the expectation function  $E(\cdot)$ . The function  $(\cdot)^*$  is the complex conjugate of a complex number. The function  $vec(\cdot)$  is a matrix vectorization function that stacks the column vectors of a matrix on top of each other to form a single column vector.

*Acronyms:* The salient algorithm acronyms used in this paper are stated as follows together with their definitions:

SD-SDS: Is a Golden code SD algorithm with fixed candidate symbol subset lengths and fixed search tree search order.

SD-SDS-Descend: Is a Golden code SD algorithm with instantaneously varying subset lengths and a search tree search order sorted based on the worst-first search strategy.

SD-SDS-Ascend: Is a Golden code SD algorithm with instantaneously varying subset lengths and a search tree search order sorted based on the best-first search strategy.

SD-SDS-ES-DNN: Is a Golden code SD-SDS algorithm with a deep learning-based early stopping search criterion.

## 2 System Model

In this paper we consider an  $N_t \times N_r$  wireless MIMO channel with the transmit and receive antenna constraints which are governed by  $N_t = 2$  and  $N_r \in [N_t: 8)$ . The Golden code wireless channel matrix for timeslot  $i$  is defined as  $\mathbf{H}_i \in \mathbb{C}^{N_r \times N_t}, \forall i \in [1: 2]$ . The wireless channel is fast frequency-flat fading which implies that the wireless channel matrix entries change for each transmission timeslot. The wireless channel matrix entries are drawn from an independent and identically distributed (i.i.d) zero-mean complex Gaussian distribution  $CN(0,1)$ . This implies that each entry's wireless channel fading gain is drawn from a Rayleigh distribution. The wireless channel matrix is assumed to be known at the receiver side.

Each Golden code super symbol is formed from a pair of M-QAM symbols that carry the  $\log_2 M$  information data bits. The way the transmission works is that the random data bit streams, at the physical layer, are packaged into 4 independent M-QAM complex symbols that each carry  $\log_2 M$  data bits. Then 2 of the 4 complex M-QAM symbols are selected to form the first Golden code super symbol, and the remaining 2 M-QAM symbols are used to form the second Golden code super symbol. In transmission timeslot 1, the first Golden code super symbol,  $x_{11}$ , is sent from transmit antenna 1, and the second Golden code super symbol,  $x_{12}$ , is sent from transmit antenna 2. In transmission timeslot 2, the M-QAM symbol pairs used to construct the Golden code super symbol 1 are used to construct the third Golden code super symbol,  $x_{21}$ , transmitted from transmit antenna 1. In the same transmission timeslot 2, a fourth Golden code super symbol is created from the same M-QAM symbol pairs used to construct the

second Golden code super symbol. The fourth Golden code super symbol,  $x_{22}$ , is transmitted in timeslot 2 from transmit antenna 2. The Golden code super symbols are constructed from the complex M-QAM symbol pairs as follows:  $x_{11} = \frac{\alpha}{\sqrt{5}}(x_1 + x_2\theta)$ ,  $x_{12} = \frac{\alpha}{\sqrt{5}}(x_3 + x_4\theta)$ ,  $x_{21} = \frac{\bar{\alpha}}{\sqrt{5}}(x_1 + x_2\bar{\theta})$ , and  $x_{22} = j\frac{\bar{\alpha}}{\sqrt{5}}(x_3 + x_4\bar{\theta})$ . The scalar parameters  $\alpha, \bar{\alpha}, \theta$  and  $\bar{\theta}$  are defined as follows:  $\alpha \triangleq 1 + j\bar{\theta}$ ,  $\bar{\alpha} \triangleq 1 + j\theta$ ,  $\theta = \frac{1+\sqrt{5}}{2}$  and  $\bar{\theta} = \frac{1-\sqrt{5}}{2}$ . The complex M-QAM symbols are  $x_1, x_2, x_3, x_4 \in \Omega_M$ , where  $\Omega_M$  is an arbitrary square M-QAM signal constellation. At the receiver side, the distorted Golden code super symbols are perturbed using the following system equation in (B.1) for timeslot  $i, \forall i \in [1: 2]$ :

$$\mathbf{y}_i = \mathbf{H}_i \mathbf{x}_i + \mathbf{n}_i \quad (\text{B.1})$$

where  $\mathbf{x}_i = [x_{i1} \ x_{i2}]^T$  is the Golden code super symbol transmission vector for timeslot  $i$ ,  $\mathbf{y}_i \in \mathbb{C}^{N_r \times 1}$  is the received perturbed Golden code super symbol signal vector for timeslot  $i$  and  $\mathbf{n}_i \in \mathbb{C}^{N_r \times 1}$  is the noise vector for timeslot  $i$ . The Golden code super symbol power is constrained to unity, i.e.  $E(|x_{i1}|^2) = E(|x_{i2}|^2) = 1$ . The noise vectors  $\mathbf{n}_i$  noise entries are drawn from an i.i.d zero-mean complex Gaussian distribution  $CN(0, \sigma^2)$  where  $\sigma^2 = \frac{N_t}{\bar{\gamma}}$ . The average received SNR per receive antenna is  $\bar{\gamma}$ . The average noise power  $\sigma^2$  is assumed to be known at the receiver side.

In our paper, we rely on the alternative representation of the system model in (B.1). As per [12] and [16], the transmission vector is based on the complex M-QAM symbols instead of the Golden code super symbols. This is achieved by rearranging the system model in (B.1) using the following rules in (B.2)

$$\mathbf{y} = \tilde{\mathbf{H}} \mathbf{u} + \mathbf{n} \quad (\text{B.2})$$

where  $\tilde{\mathbf{H}} = [\tilde{\mathbf{H}}_1 \ \tilde{\mathbf{H}}_2]^T \in \mathbb{C}^{2N_r \times 2N_t}$ ,  $\mathbf{y} = [\mathbf{y}_1 \ \mathbf{y}_2]^T \in \mathbb{C}^{2N_r \times 1}$ ,  $\mathbf{n} = [\mathbf{n}_1 \ \mathbf{n}_2]^T \in \mathbb{C}^{2N_r \times 1}$ ,

$$\mathbf{u} = [x_1 \quad x_2 \quad x_3 \quad x_4]^T, \quad \tilde{\mathbf{H}}_1 = \frac{1}{\sqrt{5}} \mathbf{H}_1 \begin{pmatrix} \alpha & \alpha\theta & 0 & 0 \\ 0 & 0 & \alpha & \alpha\theta \end{pmatrix} \text{ and } \tilde{\mathbf{H}}_2 = \frac{1}{\sqrt{5}} \mathbf{H}_2 \begin{pmatrix} \bar{\alpha} & \bar{\alpha}\bar{\theta} & 0 & 0 \\ 0 & 0 & j\bar{\alpha} & j\bar{\alpha}\bar{\theta} \end{pmatrix}.$$

The wireless channel matrix  $\tilde{\mathbf{H}}_i \in \mathbb{C}^{N_r \times 2N_t}$  is the modified wireless channel matrix, for timeslot  $i$ , that includes the Golden code super symbol constants  $\alpha, \theta, \bar{\alpha}$  and  $\bar{\theta}$ . The rest of the paper will use the system model in (B.2) for the Golden code sphere-decoding based detection algorithms.

### 3 Golden code SD-SDS Overview

The authors in [16] introduced the SD-SDS algorithm, a modified version of the SE-SD algorithm described in [12]. The SD-SDS algorithm is slightly different from the SE-SD algorithm because it does not perform the SD search over the full signal cardinality of the M-QAM constellations. In [12], the SE-SD algorithm performs the SD search over the full sorted M-QAM symbol candidates. The candidate symbols in SE-SD, just like in SD-SDS, are sorted in ascending order from the closest complex M-QAM symbol to the furthest. In SD-SDS, the furthest sorted candidate symbols are discarded as they are least likely to have been the transmitted symbols. The other difference is that the search tree search order of the SD-SDS algorithm is not sorted using the best-first search strategy as described in [12]. The SD-SDS algorithm search order execution is fixed and not sorted based on channel conditions. The SD-SDS algorithm is shown in Algorithm B.1.

#### Algorithm B.1: SD-SDS [16]

**Input:**  $\mathbf{y}, \tilde{\mathbf{H}}, \bar{\gamma}, N_t$

**Output:**  $\hat{i}_1, \hat{i}_2, \hat{i}_3, \hat{i}_4$

1. **global**  $result \leftarrow [], r^2$
2. **global**  $lattice\_point\_found\_flag$
3.  $\mathbf{z}, \tilde{\mathbf{R}}, \hat{\mathbf{x}}[ ] \leftarrow qrDecoder(\mathbf{y}, \tilde{\mathbf{H}})$
4.  $subsetList[ ] \leftarrow getSortedSubsets(\hat{\mathbf{x}}[ ], \bar{\gamma})$
5.  $\sigma^2 \leftarrow \frac{N_t}{\bar{\gamma}}, K \leftarrow 10, N \leftarrow 2N_t$
6.  $r^2 \leftarrow 2\sigma^2 KN - \mathbf{z}^H \left( \mathbf{I} - \tilde{\mathbf{R}}(\tilde{\mathbf{R}}^H \tilde{\mathbf{R}})^{-1} \tilde{\mathbf{R}}^H \right) \mathbf{z}$  [13, Eq(28)]
7.  $\mathbf{I}$  is a  $2N_t \times 2N_t$  identity matrix.
8.  $lattice\_point\_found\_flag \leftarrow False$ ,
9.  $layer \leftarrow 2N_t, dist \leftarrow 0$
10.  $sphere\_decoder \left( \begin{array}{l} \mathbf{z}, \tilde{\mathbf{R}}, subsetList[ ], \\ dist, \\ layer \end{array} \right)$
11. **if**  $lattice\_point\_found\_flag \neq False$  **then**
12.  $\hat{i}_4 \leftarrow result[4], \hat{i}_3 \leftarrow result[3], \hat{i}_2 \leftarrow result[2], \hat{i}_1 \leftarrow result[1]$

```

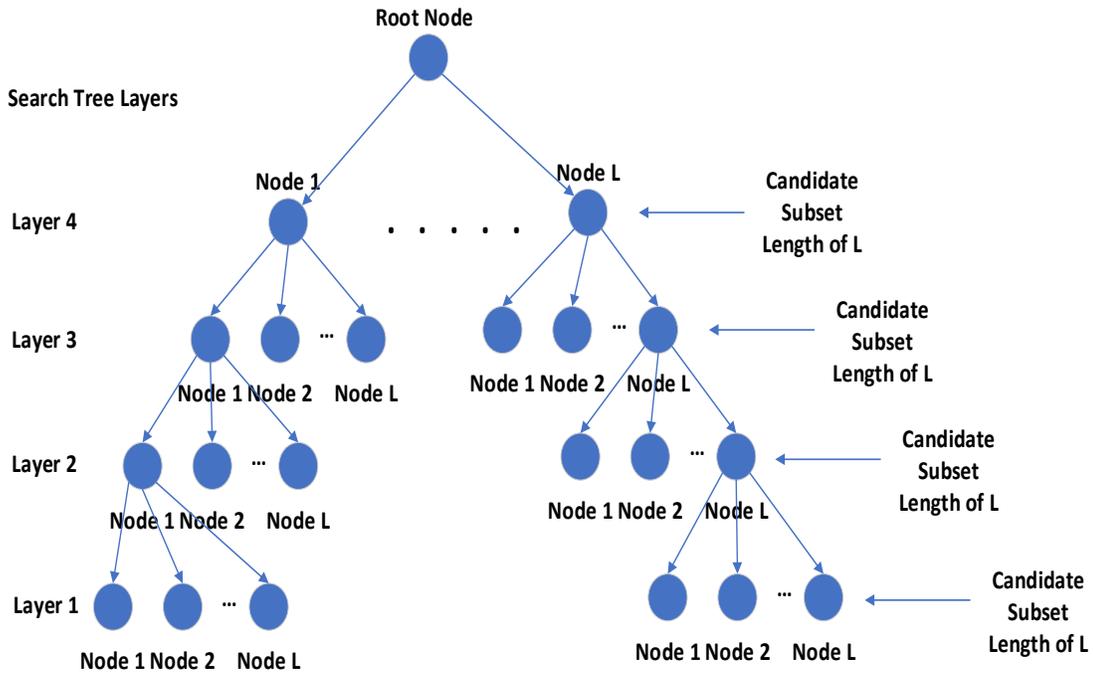
13. else
14.    $\mathbf{z}, \tilde{\mathbf{R}}, \hat{\mathbf{x}}[\ ] \leftarrow qrDecoder(\mathbf{y}, \tilde{\mathbf{H}})$ 
15.   for  $q \leftarrow 1$  to 4 do
16.      $d \leftarrow [ ]$ 
17.     for  $i \leftarrow 1$  to  $M$  do
18.        $d[i] \leftarrow |\hat{x}[q] - u_i|^2, \forall u_i \in \Omega_M$ 
19.     end for
20.      $\hat{i}_q \leftarrow \underset{i}{\operatorname{argmin}}(d[ ])$ 
21.   end for
22. end if
23. function  $qrDecoder(\mathbf{y}, \tilde{\mathbf{H}})$ 
24.    $\tilde{\mathbf{Q}}, \tilde{\mathbf{R}} \leftarrow qr(\tilde{\mathbf{H}})$ 
25.    $\mathbf{z} \leftarrow \tilde{\mathbf{Q}}^H \mathbf{y}, \hat{\mathbf{x}} \leftarrow [ ]$ 
26.   for  $q \leftarrow 4$  to 1 do
27.      $\hat{x}[q] \leftarrow \frac{(z[q] - \sum_{l=q+1}^4 \tilde{\mathbf{R}}[q][l] \hat{x}[l])}{\tilde{\mathbf{R}}[q][q]}$ 
28.   end for
29. return  $\mathbf{z}, \tilde{\mathbf{R}}, \hat{\mathbf{x}}[ ]$ 
30. function  $getSortedSubsets(\hat{\mathbf{x}}[ ], \bar{\gamma})$ 
31.    $subsetList \leftarrow [ ]$ 
32.   if  $\bar{\gamma} \leq \gamma_T^M$  then
33.      $L \leftarrow L_{low}^M$ 
34.   else
35.      $L \leftarrow L_{high}^M$ 
36.   end if
37.   for  $q \leftarrow 1$  to 4 do
38.      $dict \leftarrow [ ]$ 
39.     for  $i \leftarrow 1$  to  $M$  do
40.        $dict[i] \leftarrow |\hat{x}[q] - u_i|^2, \forall u_i \in \Omega_M$ 
41.     end for
42.      $indices[ ] \leftarrow sort(dict[ ], "ascend")$ 
43.      $subsetList[q] \leftarrow indices[1:L]$ 
44.   end for
45. return  $subsetList[ ]$ 

```

It is evident from Algorithm B.1 that the sorted subsets are generated by sorting the candidate symbols, in ascending order, using the square of the Euclidean distance between the estimated complex M-QAM symbols from the QR decoder and the exact complex M-QAM symbols from the signal constellation. The sorted symbol candidates are then truncated by taking the first  $L$  symbol indices of the sorted candidates. Each SD-SDS search layer will have its own  $L$ -dimensional sorted detection subset list for the candidate symbols since each of the M-QAM symbols in the set  $\{x_1 \ x_2 \ x_3 \ x_4\}$  are perturbed by different instantaneous fading channels

and noise conditions. The order of execution of the SD-SDS search tree search order is shown in the following one-to-one correspondence  $[\hat{x}_4 \mapsto 4, \hat{x}_3 \mapsto 3, \hat{x}_2 \mapsto 2, \hat{x}_1 \mapsto 1]$ . The correspondence shows the mapping of which symbols are estimated in each search layer, numbered 1 to 4, and which order are the layers executed in the search tree. This order does not change, unlike in the case of the SE-SD algorithm, in which the search order changes based on the instantaneous wireless channel fading power. The SD-SDS search tree searches for lattice points,  $\tilde{\mathbf{R}}\mathbf{u}$ , that lie inside the hypersphere,  $\|\mathbf{z} - \tilde{\mathbf{R}}\mathbf{u}\|_F^2 \leq r^2$ , with radius  $r$ . The vector  $\mathbf{z} = \tilde{\mathbf{Q}}^H \mathbf{y} = \tilde{\mathbf{R}}\mathbf{u} + \tilde{\mathbf{Q}}^H \mathbf{n} \in \mathbb{C}^{2N_t \times 1}$  is the received signal vector,  $\tilde{\mathbf{R}} \in \mathbb{C}^{2N_t \times 2N_t}$  is a random upper triangular matrix related to the wireless channel matrix,  $\tilde{\mathbf{H}}$ , via the reduced QR decomposition  $\tilde{\mathbf{H}} = \tilde{\mathbf{Q}}\tilde{\mathbf{R}}$ . The matrix  $\tilde{\mathbf{Q}} \in \mathbb{C}^{2N_r \times 2N_t}$  is a unitary matrix.

If no lattice point lies inside the hypersphere, the sub-optimal QR decoder is used to output the estimated transmitted symbol indices directly. If a lattice point lies inside the hypersphere, then the closest lattice point to the received signal vector is found using the Schnorr-Euchner (SE) search strategy as described in [12]. This closest lattice point contains the indices of the estimated transmitted symbols.



**Fig. B. 1 SD-SDS Search Tree.**

Looking at Fig. B.1, we see the SD-SDS search tree with each of the 4 search layers having a fixed length  $L$ -dimensional sorted candidate symbol subset shown as nodes 1 to  $L$ . The search for the closest lattice point to the received signal vector is performed using a search tree that combines the depth-first search strategy with the SE strategy. The depth-first search strategy ensures that the SD-SDS algorithm finds the closest lattice point to the received signal vector earlier in the search as possible. This is especially true for SE-SD and SD-SDS as the most likely transmitted candidate symbols are placed first in the candidate symbol subset and therefore are used first in the search on all search layers. This limits the number of lattice points found inside the hypersphere, lowering the detection complexity. The limitation occurs because after finding the first lattice point candidate inside the hypersphere, the SD-SDS adjusts the radius of the hypersphere to a smaller radius based on the distance between the lattice point and the received signal vector. It means only lattice points closer to the received signal vector will be considered going forward. Since the best candidate symbols are placed first at each search layer, it implies that at high instantaneous SNR, we can expect that the first lattice point found inside the hypersphere is the closest lattice point to the received signal vector. However, despite finding the closest lattice point, the SD-SDS algorithm continues searching all the unvisited nodes of the search tree, using the SE strategy and testing if they possibly lie inside the hypersphere. Therefore, this is an opportunity to lower the decoding latency by prematurely terminating the SD-SDS search the moment the first lattice point is found inside the hypersphere, under good instantaneous channel conditions.

The best-case scenario for the search tree is that the average SNR will be as high as possible, i.e.  $\sigma^2 \rightarrow 0$ , such that there is a very high occurrence of high instantaneous SNR channel conditions. This leads to a smaller Hypersphere radius since  $r^2 \propto \sigma^2$  [13]. The sorted candidate subsets, for each search layer, and the depth-first search strategy can be relied upon to find the closest lattice point to the received signal vector as the first lattice point inside the Hypersphere. The first lattice point exists on the far left of the search tree in Fig. B.1. This is because at high instantaneous SNR, the transmitted symbols experience minimal perturbation, and thus, any candidate symbol closest to the estimated M-QAM symbol is most likely the transmitted symbol. This makes the sorted candidate symbol subset a reliable subset containing the most likely transmitted symbols as the first symbols in the subset.

The worst-case scenario exists when the average SNR is low, i.e.  $\sigma^2 \rightarrow \infty$ , and the search radius becomes very large. A low average SNR also implies a high occurrence of low instantaneous SNR channel conditions. This makes the sorted candidate symbol subset unreliable as it no longer holds that the first symbols in the sorted candidate symbol subset are the most likely transmitted symbols. It then becomes possible that the closest lattice point to the received signal vector exists at the far right of the search tree in Fig. B.1, i.e. last lattice point. Under the worst-case scenario, it is obvious to see that the search tree detection complexity is dominated by the detection complexity at search layer 1. For the search tree to find the lattice point at the far right of the search tree, it will have to compute the Euclidean distance calculations in layer 1  $L^4$  times. For layer 2 up to layer 4, the search tree computes the Euclidean distance calculations  $L^3, L^2$ , and  $L$  times, respectively. The worst-case scenario makes the search tree equivalent to the ML detector, with detection subset length of  $L$ , as the order of execution of the search tree approaches  $O(L^{2N_t})$ .

The next Section 4 presents the proposed low complexity analytical modified SD-SDS algorithm. The proposed algorithm exploits the inherent weaknesses of the SE-SD and the SD-SDS algorithms to offer a detection algorithm with lower Golden code decoding latency compared to SD-SDS and SE-SD.

## 4 Proposed Low Complexity Analytical Sphere-Decoder

In this Section we propose a low detection complexity analytical modified Golden code SD-SDS based algorithm. This algorithm is presented in Algorithm B.2. We will explain the new concepts as we go along as we explain the workings of Algorithm B.2. Before we discuss the workings of Algorithm B.2, we will illuminate the salient differences between Algorithm B.1 and B.2.

- Since the search tree search order for Algorithm B.2 is dynamic, unlike in Algorithm B.1 where it is fixed, we need the candidate symbol subsets to follow the search tree search order of the M-QAM symbols. The function *sortSubsetOrder*( $\cdot$ ) in Algorithm B.2 makes sure that the candidate symbol subsets follow the M-QAM symbol search order. The *searchOrder*[ ] array in Algorithm B.2 is used to track the search tree search order of the M-QAM symbols.

- Algorithm B.2 sorts the wireless channel based on the dynamic search tree search order. Algorithm B.1 does not sort the wireless channel since its search tree search order is fixed. The

function  $sortChannel(\cdot)$  in Algorithm B.2 is responsible for sorting the wireless channel based on the search tree search order.

- Algorithm B.2 uses the function  $unSortOutput(\cdot)$  to restore the M-QAM symbol order to prior sorting so that the decoded output order of M-QAM symbols is predictable. Algorithm B.1 has a fixed order of M-QAM symbols output from the sphere decoder; hence it does not need this function.

Algorithm B.2 is presented below:

**Algorithm B.2: SD-SDS-Descend**

**Input:**  $\mathbf{y}, \tilde{\mathbf{H}}, \bar{\gamma}, N_t$

**Output:**  $\hat{i}_1, \hat{i}_2, \hat{i}_3, \hat{i}_4$

1. **global**  $result \leftarrow [], r^2$
2. **global**  $lattice\_point\_found\_flag$
3.  $\hat{\mathbf{x}}[] \leftarrow qrDecoder(\mathbf{y}, \tilde{\mathbf{H}})$
4.  $\sigma^2 \leftarrow \frac{N_t}{\bar{\gamma}}$
5.  $subsetList[], searchOrder[] \leftarrow getSortedSubsets(\hat{\mathbf{x}}[], \bar{\gamma}, \sigma^2)$
6.  $orderedSubsets[] \leftarrow sortSubsetOrder \left( \begin{array}{c} subsetList[], \\ searchOrder[] \end{array} \right)$
7.  $\mathbf{z}', \tilde{\mathbf{R}}' \leftarrow sortChannel(\mathbf{y}, \tilde{\mathbf{H}}, searchOrder[])$
8.  $K \leftarrow 10, N \leftarrow 2N_t$
9.  $r^2 \leftarrow 2\sigma^2KN - \mathbf{z}'^H \left( \mathbf{I} - \tilde{\mathbf{R}}' \left( \tilde{\mathbf{R}}'^H \tilde{\mathbf{R}}' \right)^{-1} \tilde{\mathbf{R}}'^H \right) \mathbf{z}'$  [13, Eq(28)]
10.  $\mathbf{I}$  is a  $2N_t \times 2N_t$  identity matrix
11.  $lattice\_point\_found\_flag \leftarrow False$ ,
12.  $layer \leftarrow 2N_t, dist \leftarrow 0$
13.  $sphere\_decoder \left( \begin{array}{c} \mathbf{z}', \tilde{\mathbf{R}}', orderedSubsets[], \\ dist, \\ layer \end{array} \right)$
14. **if**  $lattice\_point\_found\_flag \neq False$  **then**
15.  $output \leftarrow unSortOutput(result[], searchOrder[])$
16.  $\hat{i}_4 \leftarrow output[4], \hat{i}_3 \leftarrow output[3], \hat{i}_2 \leftarrow output[2], \hat{i}_1 \leftarrow output[1]$
17. **else**
18.  $\hat{\mathbf{x}}[] \leftarrow qrDecoder(\mathbf{y}, \tilde{\mathbf{H}})$
19. **for**  $q \leftarrow 1$  **to** 4 **do**
20.  $d \leftarrow []$
21. **for**  $i \leftarrow 1$  **to**  $M$  **do**
22.  $d[i] \leftarrow |\hat{x}[q] - u_i|^2, \forall u_i \in \Omega_M$
23. **end for**
24.  $\hat{i}_q \leftarrow \underset{i}{\operatorname{argmin}}(d[])$

```

25. end for
26. end if
27. function qrDecoder( $\mathbf{y}, \tilde{\mathbf{H}}$ )
28.    $\tilde{\mathbf{Q}}, \tilde{\mathbf{R}} \leftarrow qr(\tilde{\mathbf{H}})$ 
29.    $\mathbf{z} \leftarrow \tilde{\mathbf{Q}}^H \mathbf{y}, \hat{\mathbf{x}} \leftarrow []$ 
30.   for  $q \leftarrow 4$  to 1 do
31.      $\hat{x}[q] \leftarrow \frac{(z[q] - \sum_{l=q+1}^4 \tilde{\mathbf{R}}[q][l] \hat{x}[l])}{\tilde{\mathbf{R}}[q][q]}$ 
32.   end for
33. return  $\hat{\mathbf{x}}[ ]$ 
34. function getSortedSubsets( $\hat{\mathbf{x}}[ ], \bar{\gamma}, \sigma^2$ )
35.    $subsetList \leftarrow [ ], dMinList \leftarrow [ ]$ 
36.    $grad \leftarrow [2.4, 1.8, 1.6, 1.4, 1.2, 1.0, 0.8, 0.4]$ 
37.    $intercept \leftarrow [5, 4, 3, 2, 1, 0, -1, -2]$ 
38.    $order \leftarrow "descend"$ 
39.   for  $q \leftarrow 1$  to 4 do
40.      $dict \leftarrow [ ]$ 
41.     for  $i \leftarrow 1$  to  $M$  do
42.        $dict[i] \leftarrow |\hat{x}[q] - u_i|^2, \forall u_i \in \Omega_M$ 
43.     end for
44.      $indices, distances \leftarrow sort(dict[ ], "ascend")$ 
45.      $d_{min} \leftarrow distances[1]$ 
46.     if  $\bar{\gamma} \leq \gamma_T^M$  then
47.        $L \leftarrow L_{low}^M$ 
48.     else
49.        $L \leftarrow L_{high}^M$ 
50.     end if
51.      $S = L$ 
52.     for  $vv \leftarrow 1$  to 8 do
53.       if  $d_{min} < grad[vv] \sigma^2$  then
54.          $S \leftarrow \lfloor \frac{L}{2} + intercept[vv] \rfloor$ 
55.       end if
56.     end for
57.      $subsetList[q] \leftarrow indices[1:S]$ 
58.      $dMinList[q] \leftarrow d_{min}$ 
59.   end for
60.    $qList[ ] \leftarrow sort(dMinList[ ], order)$ 
61.    $searchOrder[ ] \leftarrow qList[ ]$ 
62. return  $subsetList[ ], searchOrder[ ]$ 

```

Algorithm B.2 explains the workings of the worst-first search algorithm denoted by SD-SDS-Descend. Like in Algorithm B.1, the SD-SDS-Descend algorithm determines the sub-optimal M-QAM estimates from the QR decoder. These M-QAM estimates are then used to sort the candidate M-QAM symbols, from the M-QAM constellation, in ascending order from the

closest candidate symbol to the estimated M-QAM symbol to the furthest. The candidate symbols are sorted for each M-QAM symbol estimate,  $\hat{x}_q \forall q \in [1:4]$ . The metric used for sorting in ascending order is the Euclidean distance between the estimated or noisy M-QAM symbol and the exact M-QAM symbol from the M-QAM constellation, i.e.  $d_i^q = |\hat{x}_q - u_i|^2, \forall u_i \in \Omega_M, \forall q \in [1:4], \forall i \in [1:M]$ . Now, to set the candidate symbol subset lengths based on the instantaneous channel conditions experienced by each estimated M-QAM symbol, we first define a metric that considers the instantaneous fading power and noise power. The metric is the distance between the closest M-QAM complex symbol candidate to the estimated M-QAM complex symbol, i.e.  $d_{min}^q \triangleq \min |\hat{x}_q - u_i|^2, \forall u_i \in \Omega_M, \forall q \in [1:4]$ . This minimum distance metric,  $d_{min}^q \forall q \in [1:4]$ , is defined for each estimated M-QAM symbol. We make the following assumptions to show that the distance metric,  $d_{min}^q$ , is a function of the instantaneous wireless fading power and the noise power. We will carry out the proof assuming that the previously estimated M-QAM symbols from the QR decoder are estimated without error. In the QR decoder, we estimate the M-QAM symbols in the order,  $\hat{x}_q \forall q \in [4:1], \hat{x}_4, \hat{x}_3, \dots, \hat{x}_1$ . To estimate the complex M-QAM symbol,  $\hat{x}_q$  using the QR decoder, we use the following expression in (B.3)

$$\hat{x}_q = \frac{(z_q - \sum_{l=q+1}^4 \tilde{R}_{q,l} \hat{x}_l)}{\tilde{R}_{q,q}} \quad (\text{B.3})$$

where  $z_q$  is the  $q^{th}$  scalar element in the received vector  $\mathbf{z}$  and  $\tilde{R}_{q,q}$  is the scalar element in row  $q$  and column  $q$  of the upper triangular matrix  $\tilde{\mathbf{R}}$ . It is obvious to see from (B.3) that the subsequent M-QAM symbol estimates are dependent on the previously estimated M-QAM symbols. It then follows that if the previously estimated M-QAM symbols are estimated without error, then the currently estimated M-QAM symbol has its estimation errors dependent only on its own instantaneous wireless channel and noise perturbation, as shown in (B.4)

$$\hat{x}_q = \frac{\tilde{R}_{q,q} x_q + n_q}{\tilde{R}_{q,q}} = x_q + \frac{|n_q| e^{j\theta_q}}{|\tilde{R}_{q,q}| e^{j\theta_{q,q}}}, \forall q \in [1:4] \quad (\text{B.4})$$

where  $n_q$  is the  $q^{th}$  complex scalar element of the noise vector  $\tilde{\mathbf{Q}}^H \mathbf{n}$  and  $\tilde{R}_{q,q}$  is the scalar element in row  $q$  and column  $q$  of the upper triangular matrix  $\tilde{\mathbf{R}}$ . The terms  $|n_q|e^{j\theta_q}$  and  $|\tilde{R}_{q,q}|e^{j\theta_{q,q}}$  are the complex exponential version of the terms  $n_q$  and  $\tilde{R}_{q,q}$ , respectively. The complex M-QAM symbol  $x_q$  is the exact transmitted symbol. If we take the best-case scenario, which is at high instantaneous SNR, then the closest candidate symbol to the estimated M-QAM symbol is the transmitted symbol. If we assume that the instantaneous SNR is high for all QR decoder estimated M-QAM symbols, then the metric  $d_{min}^q$  is represented mathematically in (B.5)

$$d_{min}^q = |\hat{x}_q - x_q|^2 = \frac{|n_q|^2}{|\tilde{R}_{q,q}|^2}, \forall q \in [1:4] \quad (\text{B.5})$$

From (B.5) we can see that at high instantaneous SNR, for all estimated M-QAM symbols,  $d_{min}^q = f(n_q, \tilde{R}_{q,q})$ . Therefore, the metric  $d_{min}^q$  is a function of the instantaneous noise power and the instantaneous wireless fading power. The random upper triangular matrix entries represent the wireless channel fading. For the low instantaneous SNR scenario with estimation errors in the previously estimated M-QAM symbols, the simplification in (B.5) does not apply. However, the metric  $d_{min}^q$  will still be a function of the noise and wireless channel fading. The noise will be compounded from the previously estimated M-QAM symbols via error propagation.

The metric  $d_{min}^q$  is used in Algorithm B.2 to set the instantaneously varying subset length of the candidate symbol subset for each estimated M-QAM symbol. The subset lengths,  $S_q$ , are set based on the heuristic method shown in (B.6)

$$S_q = L; \text{ if } d_{min}^q < k\sigma^2 \text{ then } S_q = \left\lfloor \frac{L}{2} + c \right\rfloor, \forall q \in [1:4] \quad (\text{B.6})$$

where  $L$  is the initial subset length which is set as  $L = 20$  when the average SNR is at most 16dB, else it is set as  $L = 30$  for an average SNR above 16dB for the case of 64-QAM. The

initial subset length values for 64-QAM are extracted from the SD-SDS algorithm in [16, Table 2]. For the case of 256-QAM, the initial subset length is set to  $L = 80$  for an average SNR, which is at most 21dB and  $L = 120$  for an average SNR above 21dB. The constants are set as follows:  $k \in \Delta = \{2.4, 1.8, 1.6, 1.4, 1.2, 1.0, 0.8, 0.4\}$  and  $c \in \beta = \{5, 4, 3, 2, 1, 0, -1, -2\}$ . The function  $\lfloor \cdot \rfloor$  returns the largest integer less than or equal to the argument. The intuition behind this method is that if the instantaneous SNR is sufficiently high, we set the instantaneous subset lengths to a very short length. If the instantaneous SNR worsens, we increase the instantaneous length of the subset. This is intuitive as we can expect that at high instantaneous SNR, we do not need to search through many candidate symbols as compared to when the instantaneous SNR is low. The gradual shortening of the instantaneous subset lengths as the instantaneous SNR increases, lowers the average decoding latency relative to the SD-SDS algorithm using fixed-length subsets that depend on average channel conditions.

Algorithm B.2 sets the instantaneous subset lengths, for each estimated M-QAM symbol, by looping through the set  $\Delta$  to set the constant  $k$  and comparing the metric  $d_{min}^q$  to each threshold level set as  $k\sigma^2$ . If the metric  $d_{min}^q$  is not less than any threshold level, then the subset length is set as  $S_q = L$  else if it falls into one of the threshold levels, then the corresponding constant  $c$  value from set  $\beta$  is used to set  $S_q = \lfloor \frac{L}{2} + c \rfloor$ . The loop index is used to extract the value of  $c$  from the set  $\beta$ .

It is obvious to see that each estimated M-QAM symbol candidate symbol subset has a possibility of having a different subset length,  $S_q$ , to the other candidate symbol subsets of the other estimated M-QAM symbols. Using this property of differing instantaneous subset lengths for each candidate symbol subset, we can easily see that it will be beneficial at low SNR to have the subset with the smallest length being used at Layer 1 of the search tree. The layer 1 detection complexity dominates the search tree complexity at low SNR. We, therefore, propose the worst-first search strategy of sorting the search order of the search tree using the descending order of the metric  $d_{min}^q$ . From Algorithm B.2 we can see that  $d_{min}$  is calculated for each value of  $q \in [1:4]$ . This means we get the following unordered set of values of the metric  $d_{min}^q$ :  $\rho = \{d_{min}^1, d_{min}^2, d_{min}^3, d_{min}^4\}$ . We then sort the set  $\rho$  in descending order to get an ordered set  $\delta =$

$\text{sort}(\rho, \text{"descend"})$ . The descend string is used to signify that we are sorting the items in descending order. For sorting items in ascending order, we use the ascend string. Let us illustrate with an example. Let us assume the unordered set  $\rho = \{d_{min}^1 = 0.5, d_{min}^2 = 2.5, d_{min}^3 = 0.15, d_{min}^4 = 0.9\}$ .

The one-to-one correspondence which shows the mapping between the metric and the estimated symbol is as follows  $[\hat{x}_4 \mapsto d_{min}^4, \hat{x}_3 \mapsto d_{min}^3, \hat{x}_2 \mapsto d_{min}^2, \hat{x}_1 \mapsto d_{min}^1]$ . Based on the SD-SDS [16] algorithm, the wireless channel matrix  $\tilde{\mathbf{H}}$  columns are unordered as follows together with the unordered M-QAM transmission vector  $\mathbf{u}$  entries:

$$\tilde{\mathbf{H}}\mathbf{u} = [\tilde{\mathbf{H}}^1 \tilde{\mathbf{H}}^2 \tilde{\mathbf{H}}^3 \tilde{\mathbf{H}}^4][x_1 \ x_2 \ x_3 \ x_4]^T \quad (\text{B.7})$$

where  $\tilde{\mathbf{H}}^q \in \mathbb{C}^{2N_r \times 1}$  are the column vectors of the wireless channel matrix from (B.2). After sorting the unordered set  $\rho$  in descending order, we get the following ordered set  $\delta = \{d_{min}^2 = 2.5, d_{min}^4 = 0.9, d_{min}^1 = 0.5, d_{min}^3 = 0.15\}$ . As we can see, the estimated M-QAM symbol with a metric with the highest value will be searched for first in the search tree. We call it the worst-first search strategy because the metric  $d_{min}^q$  in (B.5) appears as an approximate inverse of the instantaneous SNR. Therefore, a good instantaneous SNR will yield a smaller  $d_{min}^q$  value relative to a bad instantaneous SNR. Using the sorted set  $\delta$  we sort the wireless channel matrix  $\tilde{\mathbf{H}}$  column vectors and the corresponding transmission vector  $\mathbf{u}$  entries as shown in (B.8)

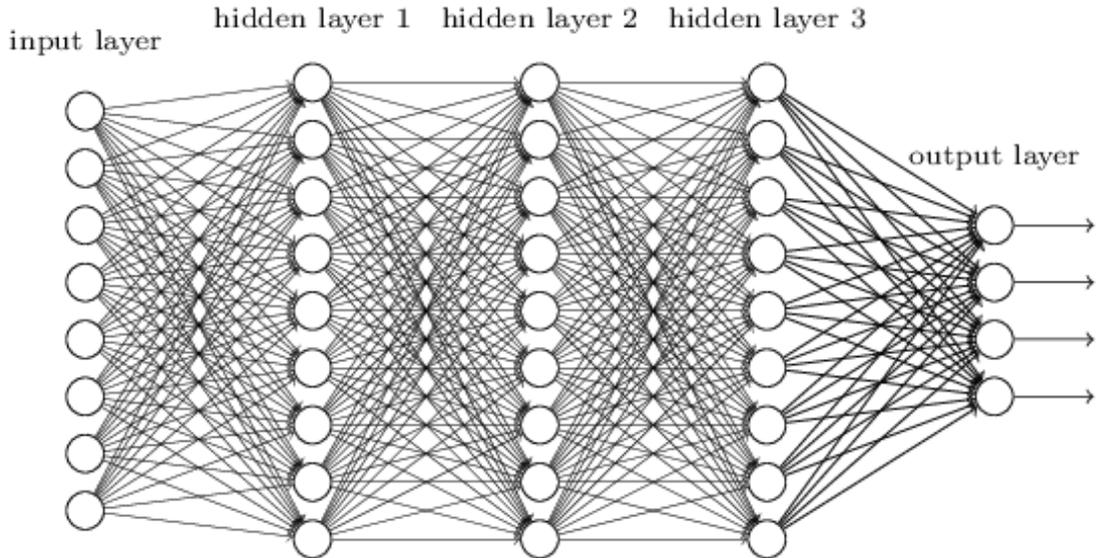
$$\tilde{\mathbf{H}}'\mathbf{u}' = [\tilde{\mathbf{H}}^3 \tilde{\mathbf{H}}^1 \tilde{\mathbf{H}}^4 \tilde{\mathbf{H}}^2][x_3 \ x_1 \ x_4 \ x_2]^T \quad (\text{B.8})$$

From (B.8) we can see that the one-to-one correspondence mapping that maps the estimated M-QAM symbols and the search layers becomes  $[\hat{x}_2 \mapsto 4, \hat{x}_4 \mapsto 3, \hat{x}_1 \mapsto 2, \hat{x}_3 \mapsto 1]$ . Search layer 1 estimates the M-QAM symbol  $\hat{x}_3$  and this is because  $\hat{x}_3$  is mapped to a metric  $d_{min}^3$  with the smallest value. Based on the heuristic method in (B.6), there is a high probability that the subset length,  $S_3$ , of the candidate symbol subset for  $\hat{x}_3$  is the smallest amongst the other estimated M-QAM candidate symbol subset lengths. For this example, the subset lengths are sorted as follows  $S_3 \leq S_1 \leq S_4 \leq S_2$ . This then shows why the worst-first search strategy is best

at the low average SNR region since the search layer 1 complexity dominates the search tree complexity. The sorted channel matrix  $\tilde{\mathbf{H}}'$  is then used to produce the modified received signal vector  $\mathbf{z}' = \tilde{\mathbf{Q}}^H \mathbf{y}$  since  $\tilde{\mathbf{H}}' = \tilde{\mathbf{Q}} \tilde{\mathbf{R}}'$ . The SD-SDS search is then performed on the sorted search tree search order to find the lattice points,  $\tilde{\mathbf{R}}' \mathbf{u}'$ , that lie inside the hypersphere  $\|\mathbf{z}' - \tilde{\mathbf{R}}' \mathbf{u}'\|_F^2 \leq r^2$ . When at least 1 lattice point is found inside the hypersphere, the *lattice\_point\_found\_flag* is set to True. The result array, *result*[], is populated with the M-QAM symbol indices derived from the most optimal lattice point found inside the hypersphere. The one-to-one correspondence [ $\hat{x}_2 \mapsto 4, \hat{x}_4 \mapsto 3, \hat{x}_1 \mapsto 2, \hat{x}_3 \mapsto 1$ ] or search order is used to unsort the SD-SDS M-QAM symbol output accordingly. If no lattice points are found inside the hypersphere, then the suboptimal QR decoder is used to determine the M-QAM symbol estimates.

## 5 Proposed Deep Learning-Based Sphere-Decoder

This Section proposes a deep learning-based SD-SDS search early stopping algorithm. The early stopping criteria of the SD-SDS search is performed on the SD-SDS Algorithm B.1. The idea is that we want to find a suitable mapping between the input and output of a deep neural network (DNN) that can predict when the SD-SDS search must be terminated prematurely. This DNN mapping will take on a structure like a typical DNN shown in Fig. B.2.



**Fig. B. 2 Typical DNN structure [21].**

In our case, we want to find a function approximator that maps the instantaneous channel conditions and noise to a binary state that determines whether early termination should take place. When early termination is deemed appropriate, it is carried out as soon as the first lattice point is found inside the hypersphere. When the first lattice point is found inside the hypersphere, the SD-SDS search tree is terminated, and we immediately output the estimated M-QAM symbol indices. There is no need to visit all the unvisited nodes in the search tree to determine if there is a closer lattice point to the received signal vector. This is because we take advantage of the fact that the candidate symbol subsets are sorted so that the most likely transmitted symbols are placed first in the candidate symbol subset. We also take advantage of the depth-first search strategy that produces lattice points quicker than the breadth-first search. The job of the DNN is to learn the channel conditions that necessitate early termination.

We define the DNN function approximator as shown in (B.9)

$$\mu \triangleq \Phi(\boldsymbol{\omega}, \boldsymbol{\tau}) \tag{B.9}$$

where  $\mu$  is the probability of initiating early termination,  $\boldsymbol{\omega}$  is the input vector of the DNN function approximator, and  $\boldsymbol{\tau}$  is the vector of the DNN model parameters that need tuning during offline training. We further define the input vector of the DNN as  $\boldsymbol{\omega} \triangleq [\hat{\mathbf{z}}^R, \hat{\mathbf{z}}^I, \tilde{\mathbf{r}}_1^R, \tilde{\mathbf{r}}_1^I] \in \mathbb{R}^{28}$  where  $\hat{\mathbf{z}}^R \triangleq (\Re(\mathbf{z}))^T$ ,  $\hat{\mathbf{z}}^I \triangleq (\Im(\mathbf{z}))^T$ ,  $\tilde{\mathbf{r}}_1^R \triangleq \text{vec}(\Re(\tilde{\mathbf{R}}))^T$  and  $\tilde{\mathbf{r}}_1^I \triangleq \text{vec}(\Im(\tilde{\mathbf{R}}))^T$ . The input vector is a 28-dimensional vector in our case since we will simulate over a  $2 \times 4$  MIMO wireless channel. For a generic  $N_t \times N_r$  MIMO wireless channel, the input vector dimension will vary depending on the number of non-zero entries found in the input vector. The input vector-only considers non-zero entries. As can be seen, the DNN uses instantaneous wireless channel fading and noise statistics to determine when early termination is suitable. The instantaneous noise statistics are indirectly catered for via the received signal vector  $\mathbf{z}$ . The instantaneous wireless channel fading is represented using the random upper triangular matrix  $\tilde{\mathbf{R}}$ , which is related to the wireless channel matrix  $\tilde{\mathbf{H}}$  via the reduced QR factorization  $\tilde{\mathbf{H}} = \tilde{\mathbf{Q}}\tilde{\mathbf{R}}$ .

The DNN function approximator  $\Phi(\cdot)$  has an architecture shown in Table B.1.

**Table B. 1: DNN Function Approximator Architecture**

Layer	Parameter Description
Input (28)	28 dimensional input vector $\omega$
(Input Layer) Dense+LeakyReLU	128 neurons. LeakyReLU slope of 0.3
BatchNormalization	void
Dropout	Probability of 0.5
(Hidden Layer 1) Dense+LeakyReLU	128 neurons. LeakyReLU slope of 0.3
BatchNormalization	void
Dropout	Probability of 0.5
(Output Layer) Dense+Sigmoid	1 neuron

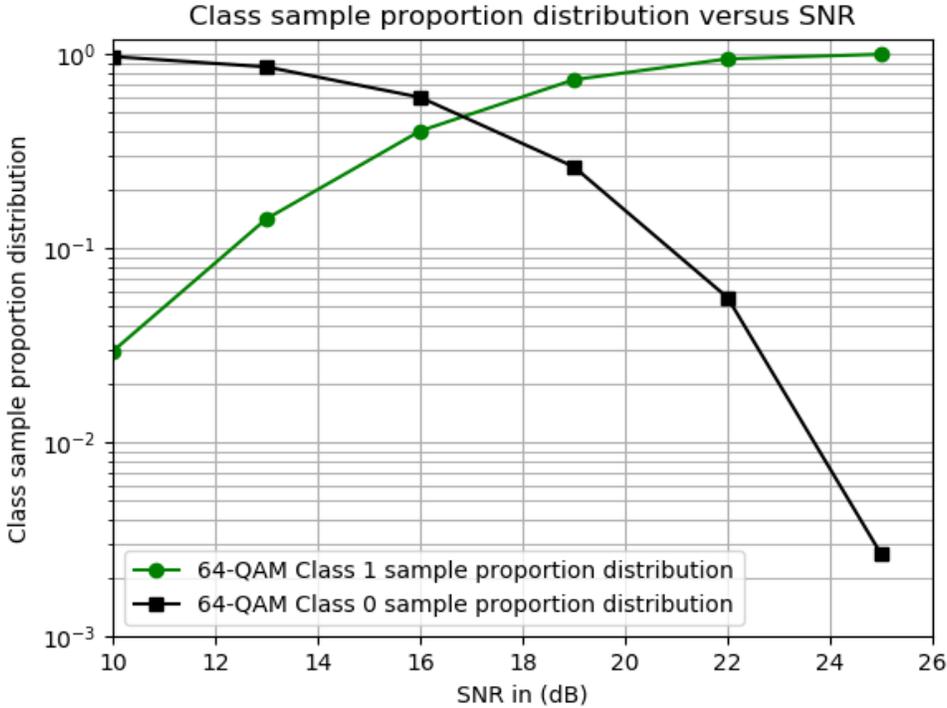
**For 64-QAM:** Batch Size=1428, Learning Rate=0.00312,  
Pseudo random seed value=70 and Maximum Epochs=2000  
**For 256-QAM:** Batch Size=8000, Learning Rate=0.0001,  
Pseudo random seed value=70 and Maximum Epochs=2000

The 64-QAM and 256-QAM DNN training learning rate, pseudo-random seed value, and batch size are determined using a meta-heuristic Genetic algorithm [22] with a fitness function dependent on the validation accuracy metric. As shown in Table B.1, the DNN architecture does not increase in size based on the M-QAM modulation order  $M$  unlike in [20]. Our architecture is thus suitable for a high-density M-QAM Golden code environment as the DNN architecture inference time will not increase as the M-QAM modulation order increases. The DNN architecture in Table B.1 is only valid for the  $2 \times 4$  MIMO wireless configuration implying that any other MIMO configuration will require the re-designing and training of a new architecture.

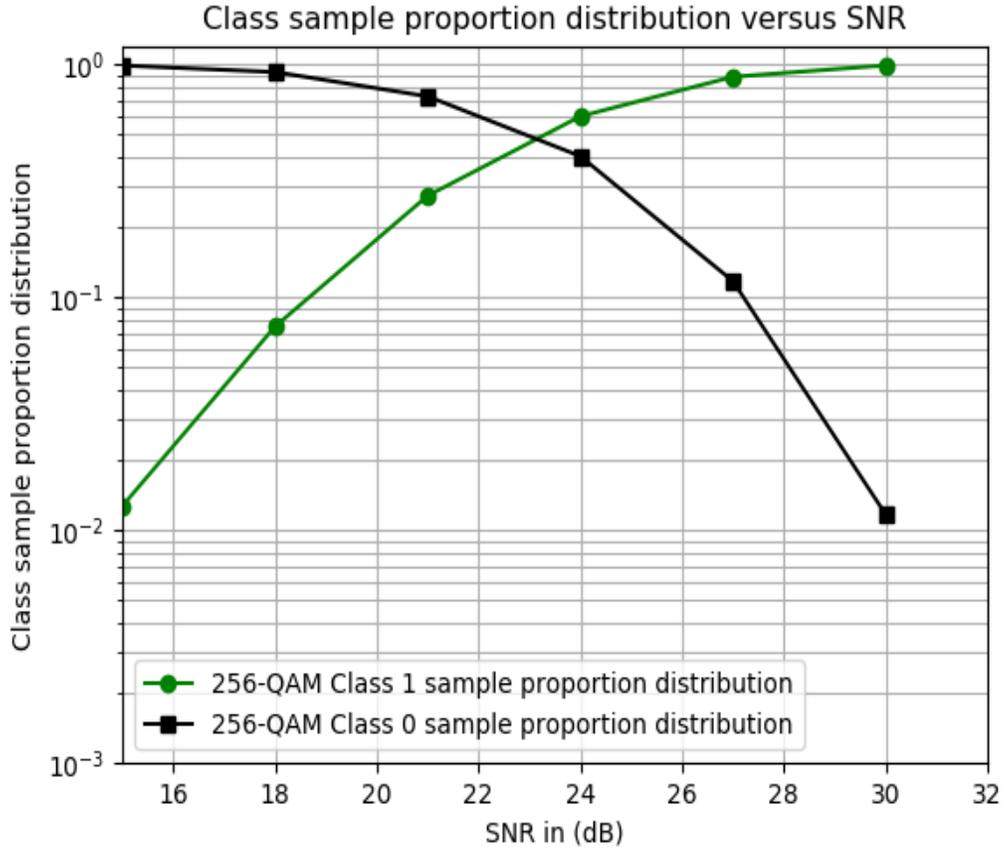
### 5.1 Offline Training of the DNN

The DNN architecture in Table B.1 is trained once offline but separately for the 64-QAM and 256-QAM scenarios. The training sample sizes are set to 20000 and 80000 for the case of 64-QAM and 256-QAM, respectively. The training samples are collected by logging the values of the instantaneous wireless fading channel and noise statistics using the definition of the input vector  $\omega \triangleq [\hat{z}^R, \hat{z}^I, \tilde{r}_1^R, \tilde{r}_1^I]$  and logging the corresponding output labels since this is supervised learning. The input vector features are each normalized into the range of [0,1] to aid good training performance by putting all features into the same scale. We prematurely terminate the SD-SDS search for each wireless channel realization when we find the first lattice point inside the hypersphere. This sole lattice point is used to deduce the estimated M-QAM symbols. If all

the estimated M-QAM symbols match the transmitted M-QAM symbols, i.e.  $\hat{x}_q = x_q \forall q \in [1: 4]$ , then the output label is set to an integer value 1 else it is set to 0. We then use this training data, with a training-to-test ratio split of 75:25, to train and evaluate the DNN architecture in Table 1. However, before training the DNN architecture in Table 1, we observe that the output labels will be unbalanced. At high SNR, we can expect most of the output labels to be 1 and at low SNR to be 0. This is because at high average SNR, the high instantaneous SNR is more frequent, and thus we can expect that the first lattice point, found inside the hypersphere, produces the correct estimates of the transmitted symbols. At low average SNR, we can expect that the high instantaneous SNR frequency is low, and hence the first lattice point found inside the hypersphere, will rarely produce reliable symbol estimates. Figs. B.3 and B.4 illuminate this observation for the distribution of output label values for the 64-QAM and 256-QAM scenarios.



**Fig. B. 3 64-QAM output label distribution.**



**Fig. B. 4 256-QAM output label distribution.**

To get the best performance from the DNN function approximator, we will need to balance the output label distribution such that the distribution of the class states 1 and 0 are close to 50:50 for all SNR values. We employ the synthetic minority over-sampling technique (SMOTE) developed by [23] to balance the output label distribution, which creates synthetic data sample points using the minority class data. The majority class data is under-sampled as per [23], leading to a class distribution ratio of 50:50 for all SNR values.

We then train the DNN architecture in Table B.1 with this balanced output label training data and use the validation accuracy metric to evaluate the performance of the DNN function approximator. The training process repeatedly feeds the DNN function approximator with the input vector training data  $\omega$ , and the DNN outputs a probability value in the range [0,1], which is then compared to the target output label data. The ADAM optimizer [24] is used to minimize

the binary cross-entropy loss function by tuning the DNN architecture model parameter weights, and biases found in the vector  $\boldsymbol{\tau}$ .

## 5.2 Online Decoding Process

The tuned model parameters in vector  $\boldsymbol{\tau}$  and the DNN architecture in Table B.1 are then saved and deployed in the simulation environment for online SD-SDS search tree early termination prediction. To determine whether early termination of the SD-SDS search is desirable, under specific wireless channel and noise conditions, we feed the trained DNN function approximator with the online input vector  $\boldsymbol{\omega}$  with normalized features. The input vector  $\boldsymbol{\omega}$  contains the instantaneous wireless channel realization. The output of the DNN will be a probability value  $\mu \in [0,1] \in \mathbb{R}$  that is then compared to a fixed threshold value  $\mu_t$ . The threshold values for 64-QAM and 256-QAM are documented in Table B.2 and are found using a heuristic approach that balances the decoding latency reduction and the BER performance at different SNR values.

**Table B. 2: 64-QAM and 256-QAM Probability thresholds**

	64-QAM Parameters		256-QAM Parameters	
Probability Thresholds	$\bar{\gamma} \leq 16 \text{ dB}$	$\bar{\gamma} > 16 \text{ dB}$	$\bar{\gamma} \leq 21 \text{ dB}$	$\bar{\gamma} > 21 \text{ dB}$
	$\mu_t = 0.2$	$\mu_t = 0.5$	$\mu_t = 0.25$	$\mu_t = 0.65$

The rule for prematurely terminating the SD-SDS search is straightforward. For every probability value  $\mu \geq \mu_t$  we prematurely terminate the SD-SDS search on the first encounter of a lattice point that lies inside the hypersphere. If the probability  $\mu < \mu_t$ , then we continue the SD-SDS search despite finding the first lattice point inside the hypersphere. This early termination technique prevents the unnecessary execution of the Euclidean distance calculations in each of the unvisited nodes in the search tree. This has the obvious effect of lowering the decoding latency of the SD-SDS search while achieving near-optimal error rate performances. The algorithm described in this section is named as SD-SDS-ES-DNN.

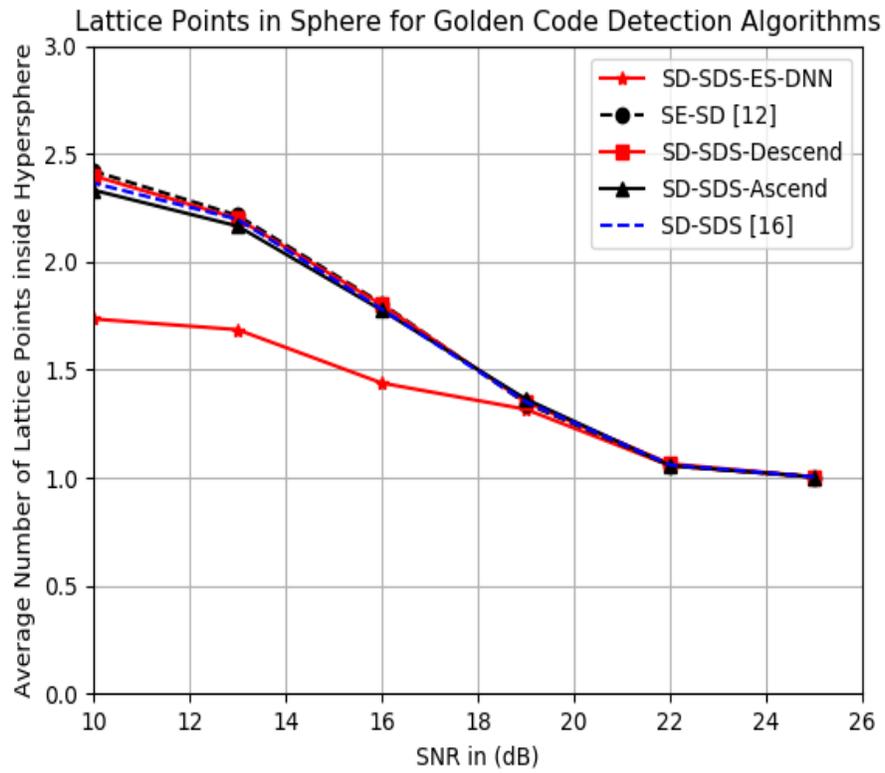
## 6 Simulation Results and Discussion

The Monte-Carlo simulation for uncorrelated wireless channels is performed for the  $2 \times 4$  wireless MIMO configuration. The number of transmit antennas in the MIMO configuration is  $N_t = 2$ , and the number of receive antennas is  $N_r = 4$ . The high-density Golden code M-QAM modulation orders considered for this simulation are the 64-QAM and 256-QAM variants. The

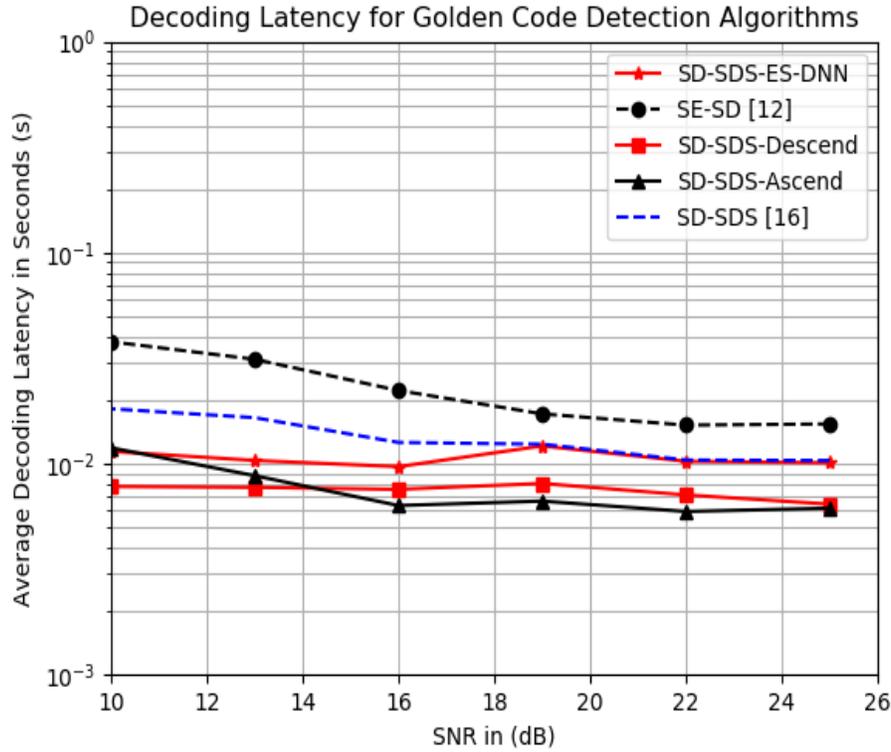
average M-QAM symbol power and Golden code super symbol power is set to 1. The Monte-Carlo simulation only shows the low complexity Golden code SD-based detection algorithms from literature versus our proposed SD-based algorithms. The performance comparison between the SE-SD [12], SD-SDS [16], and our proposed algorithms is done using the error rate performance, average decoding latency, and the average number of lattice points found inside the hypersphere. The average decoding latency is measured for each algorithm under the same computer platform.

### ***6.1 Complexity Analysis***

The detection complexity is assessed using the simulated average decoding latency and the average number of lattice points found inside the hypersphere. From Fig. B.5, we see that the SD-SDS-ES-DNN algorithm has the lowest average number of lattice points found inside the hypersphere at lower SNR. This is because the early termination algorithm terminates the SD-SDS search under good instantaneous channel conditions after finding exactly 1 lattice point inside the hypersphere. There is no difference in performance at high SNR because all the algorithms find their most optimal lattice point as the first lattice point inside the hypersphere at high instantaneous SNR. This is because the candidate symbol subset has M-QAM symbols sorted so that the most likely transmitted symbols are placed first in the subset. This coupled with the depth-first search strategy, yields a high probability, at high SNR, of finding the most optimal lattice point as the first lattice point inside the hypersphere. We must remember that the moment the SD-SDS search tree finds a lattice point closer to the received signal vector, the search tree updates the hypersphere radius to the distance of this lattice point to the received signal vector. If the first lattice point found inside the hypersphere is the most optimal or closest lattice, no other lattice points will be found inside the hypersphere.



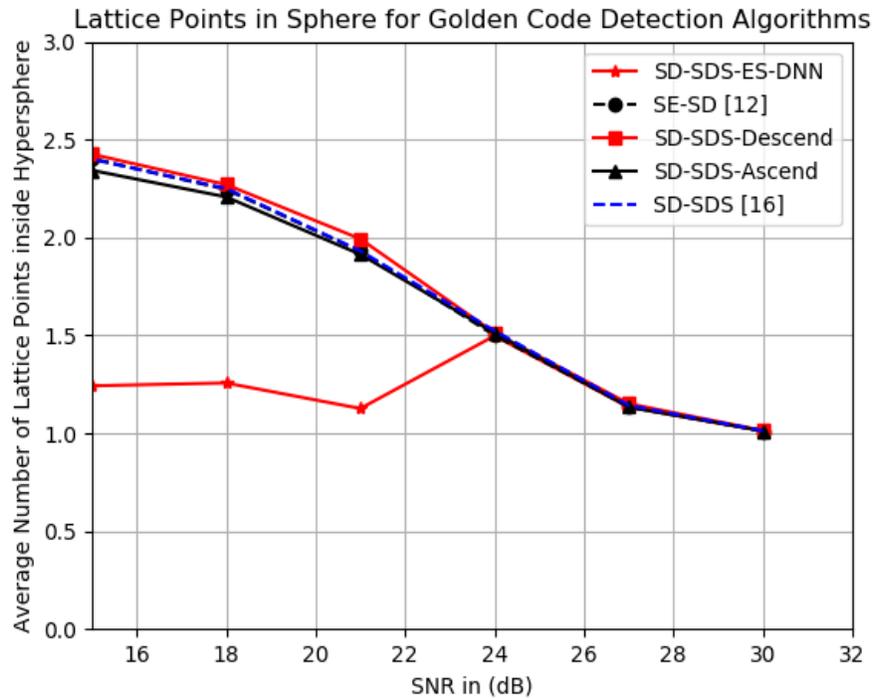
**Fig. B. 5 64-QAM Golden code SD-based detection algorithms average number of lattice points inside hypersphere for  $2 \times 4$  MIMO.**



**Fig. B. 6 64-QAM Golden code SD-based detection algorithms average decoding latency for  $2 \times 4$  MIMO.**

Fig. B.6 exhibits the Golden code SD-based detection algorithms' average decoding latency. At low SNR, we observe that the SD-SDS-Descend algorithm, which is the proposed Algorithm B.2 in this paper, has the lowest average decoding latency with at most 57% reduction in decoding latency relative to the SD-SDS from literature. At high SNR, the SD-SDS-Descend and SD-SDS-Ascend algorithms produce similar decoding latency reduction of at most 40% relative to SD-SDS. At high SNR, the source of decoding latency reduction is the smaller instantaneous subset lengths generated by the heuristic method in Eq.(B.6) of this paper. This is because the SD-SDS-Descend and SD-SDS-Ascend algorithms use the instantaneous subset lengths, which will be in the range  $S_q \in [13: 20]$  for 64-QAM at high SNR. Despite the narrow instantaneous subset length range, it is approximately 66% of the subset length used by the SD-SDS algorithm, which uses fixed-length subsets of 30 candidate symbols for 64-QAM at high SNR. Because at high instantaneous SNR the instantaneous subset lengths are shorter than that required at high average SNR by the SD-SDS with fixed-length subsets, the search tree search breadth is shortened and thus decoding latency is lowered. The BER performance is not affected

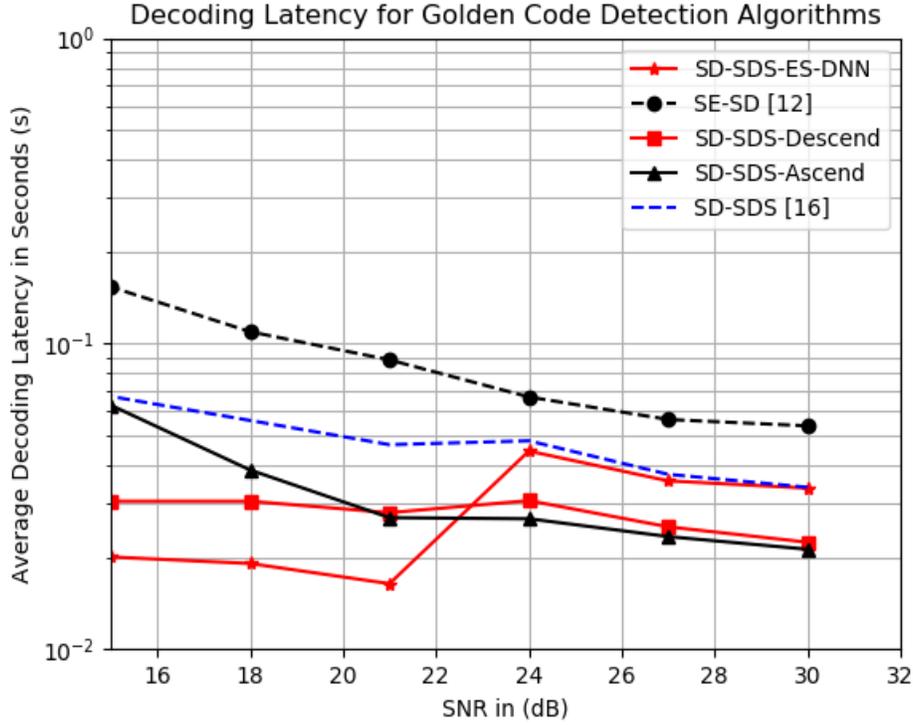
because this shortening of subset lengths only occurs at sufficiently high instantaneous SNR as per Eq.(B.6). The SD-SDS-Ascend algorithm is just the reverse of the SD-SDS-Descend because the search tree search order is sorted in ascending order using the defined metric in this paper. Despite the SD-SDS-ES-DNN having the lowest average number of lattice points inside the hypersphere, at low SNR, it has a greater decoding latency than the SD-SDS-Descend algorithm because of the effect of its DNN architecture inference time. The SD-SDS-Descend algorithm has lower decoding latency at low SNR than the other analytical decoding algorithms because it orders the search tree search order such that search tree layer 1 has the smallest candidate symbol subset. The search tree layer 1 dominates the search tree complexity at low SNR, therefore, assigning it a subset with the smallest length lowers the decoding latency of the search tree. The SE-SD algorithm has the worst decoding latency because it uses the full signal cardinality of the M-QAM constellation to search for the optimal solution. The other algorithms use candidate symbol subsets shorter than the M-QAM signal constellation cardinality. The SD algorithms are known to have their decoding complexity dependent on the search signal cardinality and search depth of the search tree [14].



**Fig. B. 7 256-QAM Golden code SD-based detection algorithms average number of lattice points inside hypersphere for  $2 \times 4$  MIMO.**

Fig. B.7 shows that the SD-SDS-ES-DNN algorithm continues to deliver the lowest average number of lattice points inside the hypersphere for the case of 256-QAM at low SNR. The reasons advanced are identical to those ventilated for the case of 64-QAM baseband modulation in Fig. B.5. However, at high SNR, the SD-SDS-ES-DNN algorithm has the same performance as the analytical SD algorithms. This is because, at high SNR, the SD-SDS-ES-DNN is virtually not prematurely terminating the SD-SDS search because of the error rate performance sensitivity at high SNR. The SD-SDS-ES-DNN algorithm relies on a DNN output probability to activate the early termination based on the wireless channel quality. Because of the inevitable prediction errors present in the DNN output, there are times when the DNN erroneously outputs a probability that enables premature termination of the SD-SDS search in unfavorable channel conditions. This will negatively impact the error rate performance at high SNR. To counter this, the SD-SDS-ES-DNN algorithm infrequently prematurely terminates the SD-SDS search to maintain the near-optimal error rate performance at high SNR.

Fig. B.8 exhibits that the proposed SD-SDS-Descend algorithm has the lowest decoding latency among the analytical SD-based algorithms, at low SNR, for the same reasons advanced in the case of 64-QAM modulation in Fig. B.6.



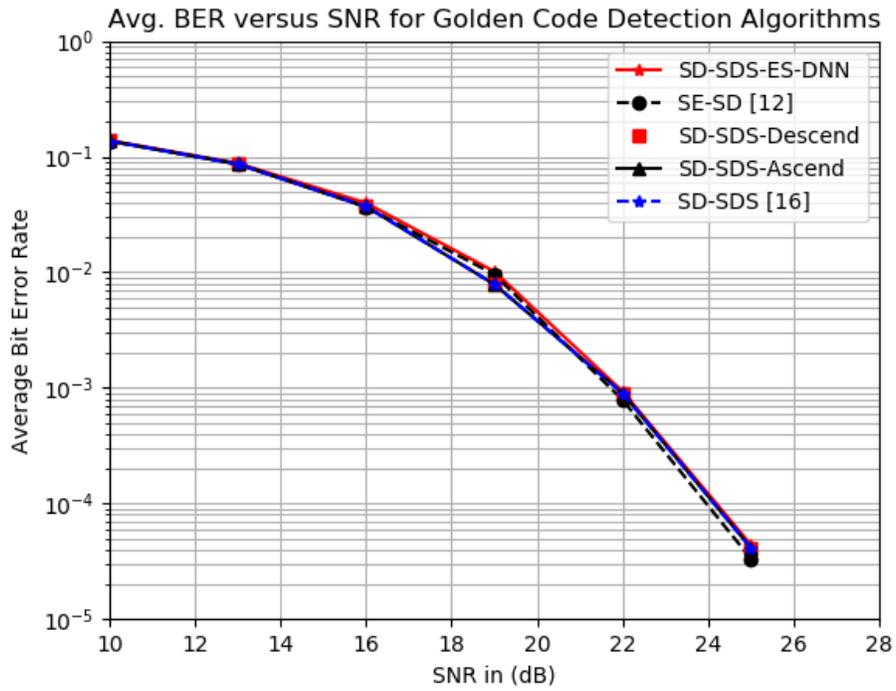
**Fig. B. 8 256-QAM Golden code SD-based detection algorithms average decoding latency for  $2 \times 4$  MIMO.**

The difference here is that for the 256-QAM case, the SD-SDS-ES-DNN algorithm has the lowest decoding latency, at low SNR, compared to all other algorithms. A decoding latency reduction of 70% is achieved relative to the SD-SDS algorithm at low SNR. This is because, for 256-QAM, the analytical SD-based algorithms visit all the unvisited nodes in the large search tree. The 256-QAM search tree is larger than the 64-QAM tree because the signal cardinality or search breadth is larger for 256-QAM. Whether the instantaneous SNR is good or not, the analytical SD-based decoding algorithms visit unvisited tree nodes to determine if a more optimal lattice point can be found inside the hypersphere. The SD-SDS-ES-DNN algorithm prematurely terminates the search when it finds 1 lattice point inside the hypersphere under good instantaneous SNR conditions. This lowers the decoding latency. Over and above this, for the 256-QAM case, the DNN architecture remains the same as that for the 64-QAM case. This implies that the DNN inference time has a marginal effect on the decoding latency for the case of 256-QAM. At high SNR, the SD-SDS-ES-DNN algorithm has a decoding latency that matches the SD-SDS algorithm decoding latency. This is because, at high SNR, the SD-SDS-

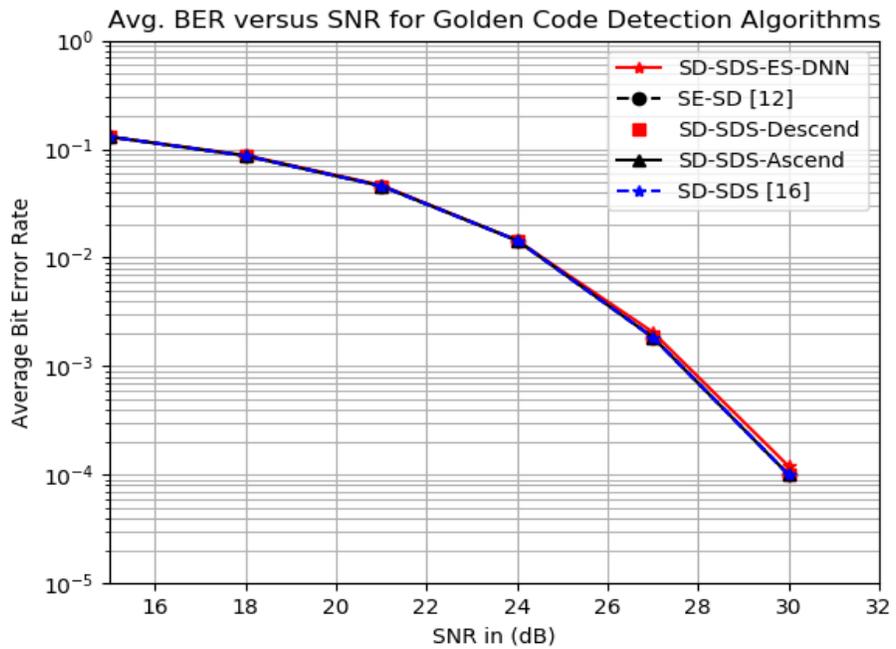
ES-DNN algorithm performs virtually no early termination of the SD-SDS search. After all, it needs to maintain a near-optimal BER performance at the expense of increased decoding latency. At high SNR, the SD-SDS-Descend and SD-SDS-Ascend algorithms achieve approximately 37% decoding latency reduction relative to the SD-SDS algorithm from the literature. This is because the SD-SDS-Descend and SD-SDS-Ascend algorithms use the instantaneous subset lengths in the range  $S_q \in [58: 65]$  for 256-QAM at high SNR. Despite the narrow instantaneous subset length range, it is approximately 50% of the subset length used by the SD-SDS algorithm, which uses fixed-length subsets of 120 candidate symbols for 256-QAM at high SNR. Because at high instantaneous SNR the instantaneous subset lengths are shorter than that required at high average SNR by the SD-SDS with fixed-length subsets, the search tree search breadth is shortened and thus decoding latency is lowered. The BER performance is not affected because this shortening of subset lengths only occurs at sufficiently high instantaneous SNR as per Eq.(B.6). The SE-SD [12] algorithm still exhibits the worst decoding latency performance relative to the proposed algorithms and the SD-SDS [16] algorithm.

## ***6.2 Error Rate Performance***

Figs. B.9 and B.10 show that the proposed SD-SDS-Descend and SD-SDS-ES-DNN algorithms achieve near-optimal BER performances despite reducing decoding latency relative to the state-of-the-art low complexity Golden code detection algorithms SE-SD [12] and SD-SDS [16].



**Fig. B. 9 64-QAM Golden code SD-based detection algorithms error rate performance for  $2 \times 4$  MIMO.**



**Fig. B. 10 256-QAM Golden code SD-based detection algorithms error rate performance for  $2 \times 4$  MIMO.**

## 7 Conclusion

In this paper, we successfully proposed a more appropriate channel quality metric to sort the SD-SDS search tree search order. The channel quality metric considered both the instantaneous wireless channel fading power and the instantaneous noise power. The SE-SD search order sorting metric, in literature, considered only the instantaneous wireless channel fading power. This is not an accurate assessment of channel quality as noise statistics dominate the SNR performance at low SNR instead of fading. We also proposed instantaneously varying candidate symbol subset lengths per search layer. The candidate symbol subset lengths varied with the instantaneous channel conditions for each estimated M-QAM symbol and allowed the search tree search order to be sorted based on the subset lengths. This led to the proposal of the worst-first search strategy, which was employed by the detection algorithm SD-SDS-Descend. The SD-based search trees have their detection complexity dominated by the search layer 1 detection complexity at low SNR. The worst-first search strategy ensured that the candidate symbol subset with the smallest subset length always got assigned to search layer 1. This assisted the SD-SDS-Descend algorithm to achieve a reduction in decoding latency of 57% relative to the SD-SDS algorithm for the case of 64-QAM modulation at low SNR. The paper also proposed a deep learning-based early termination algorithm, i.e. SD-SDS-ES-DNN, for low complexity SD-SDS small MIMO. For 256-QAM, the SD-SDS-ES-DNN algorithm achieved 70% reduction in decoding latency at low SNR relative to the SD-SDS algorithm proposed in the literature. The SD-SDS-Descend algorithm achieved 40% and 37% decoding latency reduction relative to SD-SDS, at high SNR, for the case of 64-QAM and 256-QAM, respectively. All these gains were shown to be achieved without losing any error rate performance relative to the near-optimal BER performances of SE-SD and SD-SDS.

## References

- [1] L. Zheng and D. N. C. Tse, "Diversity and Multiplexing: A Fundamental Trade-off in Multiple-Antenna Channels," *IEEE Trans. Inform. Theory*, vol. 49, no. 5, pp. 1073-1096, May 2003.
- [2] J. Belfiore, G. Rekaya and E. Viterbo, "The golden code: a 2 x 2 full-rate space-time block code with non-vanishing determinants", *International Symposium on Information Theory, ISIT 2004. Proceedings.*, Chicago, IL, 2004, pp. 310-310, 2004.

- [3] V. Torakh, H. Jafarkhani, A. R. Calderbank, "Space-Time Block Codes from Orthogonal Designs", IEEE Transactions on Information Theory, vol. 45, no 5, July 1999.
- [4] M. O. Sinnokrot and J. R. Barry, "Fast maximum-likelihood decoding of the Golden code", IEEE Trans. Wireless Commun., vol. 9, no. 1, pp. 26-31, 2010.
- [5] S. M. Alamouti, "A simple transmit diversity technique for wireless communications", IEEE Journ. Sel. Areas Commun., vol. 16, no. 8, pp.1451-1458, 1998.
- [6] H. Xu, K. Govindasamy, and N. Pillay, "Uncoded space-time labelling diversity", IEEE Commun. Lett., vol. 20, no. 8, pp. 1511-1514, 2016.
- [7] F. Riera-Palou and G. Femenias, "Improving STBC Performance in IEEE 802.11n Using Group-Orthogonal Frequency Diversity", IEEE Wireless Communications and Networking Conference, Las Vegas, NV, pp. 193-198, 2008.
- [8] N. Sharma, "Space Time Block Code for Next Generation Multi-user MIMO System", 9th International Conference on Future Networks and Communications, Procedia Computer Science, pp. 172-179, 2014.
- [9] IEEE 802.16e-2005: IEEE Standard for Local and Metropolitan Area Network - Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems - Amendment 2: Physical Layer and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands, Feb. 2006.
- [10] S. Kahraman and M. E. Çelebi, "Dimensionality reduced decoding for the golden code with the worst-case complexity of  $O(m^{1.5})$  for low range of SNR," IEEE Wireless Communications and Networking Conference (WCNC), 2012, pp. 246-250.
- [11]. S. Sirinaunpiboon, A. R. Calderbank, and S. D. Howard, "Fast essentially maximum likelihood decoding of the Golden code", IEEE Trans. Inf.Theory, vol. 57, no. 6, pp. 3537-3541, 2011.
- [12]. L. Zhang, B. Li, T. Yuan, X. Zhang, and D. Yang, "Golden code with low complexity sphere decoder", in Proc. 18th Int. Symp. Pers. Indoor Mobile Radio Commun., pp. 1-5, 2007.
- [13] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel", IEEE Trans. on Commun., vol. 51, no. 3, pp. 389-399, 2003.
- [14] J. Jaldén and B. Ottersten, "On the complexity of sphere decoding in digital communications", IEEE Trans. Signal Process., vol. 53, no. 4, pp. 1474-1484, 2005.

- [15] H. Xu and N. Pillay, "Reduced Complexity Detection Schemes for Golden Code Systems", *IEEE Access*, vol. 7, pp. 139140-139149, 2019.
- [16]. H. Xu and N. Pillay, "Multiple Complex Symbol Golden Code", *IEEE Access*, vol. 8, pp. 103576-103584, 2020.
- [17] A. Askri and G. R. Othman, "DNN assisted Sphere Decoder", 2019 *IEEE Int. Symposium on Inf. Theory (ISIT)*, Paris, France, pp. 1172-1176, 2019.
- [18] M. Mohammadkarimi, M. Mehrabi, M. Ardakani and Y. Jing, "Deep Learning-Based Sphere Decoding", *IEEE Trans. on Wireless Commun.*, vol. 18, no. 9, pp. 4368-4378, 2019.
- [19] N. T. Nguyen, K. Lee and H. Dai, "Application of Deep Learning to Sphere Decoding for Large MIMO Systems," *IEEE Transactions on Wireless Communications*, doi: 10.1109/TWC.2021.3076527.
- [20]. D. Weon and K. Lee, "Learning-Aided Deep Path Prediction for Sphere Decoding in Large MIMO Systems," *IEEE Access*, vol. 8, pp. 70870-70877, 2020.
- [21] M.A. Nielsen, "Neural Networks and Deep Learning", Determination Press, 2015.
- [22] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning", Reading, MA, USA: Addison-Wesley, 1989.
- [23] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique", *Journal of Artificial Intelligence Research*, Vol. 16, 2002.
- [24] D. B. J. Kingma, "Adam, a method for stochastic optimization", 2014.

**Paper C**

**Low Complexity Deep Learning-Assisted Golden Code Sphere-  
Decoding with Sorted Detection Subsets**

Bhekisizwe Mthethwa and Hongjun Xu

SAIEE Africa Research Journal, vol. 113, no. 2,  
June 2022.

## Abstract

Golden code is a space-time block coding (STBC) scheme that has spatial multiplexing gain over the Alamouti STBC which is widely used in modern wireless communication standards. Golden code has not been widely adopted in modern wireless standards because of its inherent high detection complexity. However, detection algorithms like the sphere-decoding with sorted detection subsets (SD-SDS) have been developed to lower this detection complexity. Literature indicates that the SD-SDS algorithm has lower detection complexity relative to the traditional sphere-decoding (SD) algorithm, for all signal-to-noise ratio (SNR) values. The SD-SDS algorithm exhibits low detection complexity at high SNR; however, at low SNR the detection complexity is higher. We propose a deep neural network (DNN) aided SD-SDS algorithm (SD-SDS-DNN) that will lower the Golden code's SD-SDS low SNR detection complexity, whilst maintaining the bit-error-rate (BER) performance. The proposed SD-SDS-DNN is shown to achieve a 75% reduction in detection complexity relative to SD-SDS at low SNR values for 16-QAM, whilst maintaining the BER performance. For 64-QAM, the SD-SDS-DNN achieves 99% reduction in detection complexity relative to the SD-SDS at low SNR, whilst maintaining the BER performance. The SD-SDS-DNN has also shown to achieve low detection complexity comparable to that of the Alamouti linear maximum likelihood (ML) detector for a spectral efficiency of 8 bits/s/Hz. For a spectral efficiency of 12 bits/s/Hz, the SD-SDS-DNN achieves a detection complexity that is 90% lower than the Alamouti linear ML detector.

# 1 Introduction

WITH the ever-increasing demand for higher data transmission rates and link reliability for wireless radio access networks (RAN), because of increasing real-time and mission-critical applications, it becomes necessary to research and develop wireless schemes that can provide high spectral efficiency, good link reliability, and low detection complexity. Multiple-input multiple-output (MIMO) wireless techniques can deliver on these requirements through space-diversity and spatial multiplexing. Space-time block coding (STBC) schemes such as Alamouti [1], uncoded space-time labeling diversity (USTLD) [2], and Golden code [3] provide time-diversity over and above space diversity. These schemes further improve wireless link reliability over MIMO wireless channels. Of the three STBC schemes, Golden code is the only full-rate, full-diversity wireless scheme, whilst the USTLD and Alamouti are half-rate and full-diversity schemes. This paper's coding rate is defined as the number of M-QAM symbols transmitted per transmit antenna per transmission timeslot. The advantage of the Golden code is that it offers full spatial-multiplexing gain relative to the Alamouti and USTLD STBC schemes. However, Alamouti STBC has the advantage of having much lower linear detection complexity relative to the Golden code STBC scheme under block-fading wireless channels. The Alamouti optimal linear decoder has order  $\vartheta(M^1)$  detection complexity, in block-fading channels [1], relative to the Golden code optimal detector with  $\vartheta(M^4)$  detection complexity. The variable  $M$  is defined as the M-QAM signal modulation order. The USTLD STBC has a coding gain advantage over the Alamouti STBC whilst achieving the same rate and diversity order [2]. However, USTLD STBC has a detection complexity of order  $\vartheta(M^2)$  since it uses joint maximum likelihood (ML) detection to decode the two transmitted symbols [2]. This USTLD STBC decoding complexity is higher than the Alamouti STBC linear detection complexity in block-fading channels. The Alamouti STBC linear decoder is shown to under-perform in terms of bit-error-rate (BER) in fast-fading channels [4] due to inter-symbol interference (ISI). The optimal detector for the Alamouti STBC scheme, in fast-fading channels, is shown in [4] to be the joint ML detector with order  $\vartheta(M^2)$  detection complexity.

Golden code STBC is a promising wireless scheme which is already incorporated into the WiMAX IEEE 802.16e standard [5]. Golden code offers spatial multiplexing gain relative to the Alamouti STBC scheme, at the expense of higher detection complexity. In modern wireless

communication systems, the following standards incorporate the Alamouti STBC scheme namely, the long-term evolution (LTE) 3GPP standard [6], wireless fidelity (WiFi) IEEE 802.11n [7] and IEEE 802.11ah low power WiFi [8]. This makes research into the reduction of the Golden code detection complexity interesting and relevant to modern wireless communication systems. The practical application of Golden code is limited due to its high detection complexity despite its advantages of spatial multiplexity gain over the Alamouti and USTLD STBC schemes. Thus, to extract the benefits of Golden code in a practical use case, researchers have embarked on developing various detection schemes to lower the Golden code's detection complexity. Therefore, this paper particularly concentrates on the Golden code STBC due to its disadvantage of being a high detection complexity scheme. The high detection complexity has a negative implication of increasing telecommunications operator base-station processing power consumption, including that of the end-users. High complexity detection schemes may also increase end-to-end link latency if powerful detection processors are not used. An increase in link latency has detrimental effects on real-time low latency applications. Our research aims to reduce this detection complexity.

In literature, [9] develops an efficient ML detection scheme that can reduce the Golden code's detection complexity to approximately  $\vartheta(M^{2.5})$ . Using dimensionality reduction of the search tree in sphere-decoding (SD), [10] manages to reduce the detection complexity to approximately  $\vartheta(M^{1.5})$ . However, [10]'s side effect is that the BER performance suffers a 1dB SNR loss compared to optimal ML detection. The fast-essentially maximum likelihood (FML) detection is developed in [11] with a detection complexity of  $\vartheta(M^2)$ . FML proves to be computationally intensive at higher modulation orders. In literature, a near-optimal detection algorithm called SD, with detection complexity of  $\vartheta(M^2)$ , is modified in [12] by minimizing the search depth, to reduce detection complexity, using the Schnorr-Euchner strategy. It is known from literature that SD detection complexity relies on the signal modulation order and the search depth [13]. The authors in [14] further optimize the FML and SD algorithms by reducing the signal cardinality by creating detection subsets. This can further reduce the detection complexity whilst maintaining the BER performance close to that achieved by FML and SD. In [15], the authors propose SD-SDS, which has low detection complexity for Golden code at high SNR with an increasingly high detection complexity as the SNR approaches 0dB. However, it has detection

complexity which is at least 1 order lower than the sphere-decoding detection subset algorithm (SD-DS) developed in [14].

Part of the challenge in [15] is that the SD initial radius is fixed per average SNR, thus at lower average SNR values, the initial radius is larger which causes a selection of many signal candidates under good instantaneous SNR conditions. This creates a high detection complexity at lower SNR values and suggests that we may need an SD initial radius calculated using the instantaneous channel conditions instead of average channel conditions. In [16], the authors develop a deep learning-based initial radius predictor that predicts an initial radius based on instantaneous channel conditions. This approach lowers the detection complexity of MIMO SD detection whilst maintaining the BER performance. Another interesting MIMO SD technique is developed by [17]. The initial radius of SD is selected and fed into a deep neural network that predicts the number of lattice points inside the hypersphere. If the predicted number of lattice points is high, then the initial radius is adjusted downwards and re-fed into the deep neural network. This is done iteratively until the number of predicted lattice points is low, at which point SD is performed with this lower initial radius that is predicted to yield a small number of lattice points inside the hypersphere. This technique yields lower detection complexity for SD. However, in our experiments, we found that [16] yields better performance than [17]. In [18], the authors propose a deep learning (DL)-aided SD for large MIMO detection. Because SD detection for large MIMO has a prohibitive computational complexity, the DL-aided SD generates a highly reliable initial candidate to accelerate the SD search for the transmitted symbols. The DL-aided SD is beneficial both from an offline training phase and online application relative to the DL-aided SD in literature. In [19], a neural network is proposed that predicts the minimum path metrics of subtrees of a SD and these predicted minimum path metrics are used for early termination in the SD search for candidates. The scheme shows significant computational complexity reduction relative to the conventional SD scheme for large MIMO, whilst exhibiting a BER performance close to the optimal detector.

Based on the literature review, we are motivated to lower the detection complexity of the Golden code SD-SDS decoder, at low SNR, for the traditional MIMO architecture. This reduction of low SNR Golden code detection complexity is important for low power wireless

communications. The algorithms in [16-19] perform deep learning-aided complexity reduction in the conventional SD algorithm, for large MIMO (*i. e.*  $N_t \geq 8$ ), except for the Golden code context with a small number of transmit antennas (*i. e.*  $N_t = 2$ ). Therefore, no literature has performed complexity reduction of the Golden code specific SD-SDS detection algorithm. The reduction in detection complexity is performed to give Golden code an edge over the Alamouti STBC scheme which is already implemented in modern wireless communication standards. Golden code has greater spectral efficiency, for the same link reliability, relative to the Alamouti STBC but at the expense of higher detection complexity which prevents it from being incorporated into broader wireless standards as an STBC scheme of choice. By embarking on lowering the detection complexity of Golden code SD-SDS, at low SNR, our paper makes the following contributions:

- (i) The SD initial hypersphere radius computation is well discussed in [20-23]. However, the computation in literature is performed for a single timeslot whereas Golden code is a 2-timeslot STBC scheme. In this paper, we derive the 2-timeslot Golden code SD-SDS [15] fixed initial radius and show that this SD-SDS has an initial radius dependent only on a single timeslot, which makes the formulas discussed in [20-23] also relevant for our use case.
- (ii) We present a modified version of the low complexity deep learning-based algorithm in [16]. This modified algorithm lowers the SD-SDS detection complexity at low SNR. The reason for the modification is because the algorithm in [16] is developed for a single timeslot scheme, whereas Golden code is a 2-timeslot STBC system. This has the effect of changing the DNN input vector length and thus requires us to design and train a new DNN architecture for the radius prediction. The algorithm in [16] is also developed for a very high detection complexity large MIMO ( $N_t \geq 10$ ) SD environment whereas our scheme needs to work for a lower detection complexity traditional MIMO ( $N_t = 2$ ) SD-SDS environment. Because of the lower complexity traditional MIMO SD-SDS environment, we only predict one radius at the output of the DNN unlike in [16]. To counter the error in prediction accuracy of a single radius prediction, we use the reverse of the approach in [17] to determine the subsequent radius predictions. This DNN algorithm is named as the SD-SDS-Radius-DNN.

- (iii) SD based algorithms are generally more complex than the sub-optimal QR decomposition detector. However, under good instantaneous channel and noise conditions, the sub-optimal QR decomposition detector produces M-QAM symbol estimates that are reliable. We, therefore, propose a novel DNN channel state predictor that uses the instantaneous channel conditions and received signal vectors to predict whether the low complexity sub-optimal QR decomposition detector estimates are good enough to be used as the actual transmitted symbols without performing the more complex SD based search.
- (iv) We also propose a low complexity detection algorithm called the SD-SDS-DNN. The proposed SD-SDS-DNN algorithm combines the high SNR low complexity benefits of the SD-SDS detector from [15], the low SNR low complexity benefits of the SD-SDS-Radius-DNN detector, and the benefits of the proposed novel DNN channel state predictor that selects between the very low complexity QR decomposition detector output and search using the SD-SDS-Radius-DNN detector.
- (v) We perform the DNN architecture designs and training for the two DNNs in the paper and perform Monte-Carlo simulations of the BER and complexity analysis of the Golden code detection algorithms discussed in this paper.

The remainder of this paper is organized as follows: Section 2, the system model of the paper is presented. In Section 3, we present the theoretical overview of SD-SDS. In Section 4, we deal with the derivation of the Golden code 2-timeslot SD initial radius. In Section 5, we present the SD-SDS deep learning algorithms. In Section 6, we perform the complexity analysis of the Golden code detection algorithms relative to SD-SDS. Section 7 presents the simulation results and discussion. Section 8 concludes the paper.

*Notation:* Bold lowercase letters are used for vectors and bold uppercase for matrices.  $(\cdot)^T$   $(\cdot)^H$ ,  $|\cdot|$ ,  $\|\cdot\|$  and  $\|\cdot\|_F$  represent the Transpose, Hermitian, Absolute Value, Euclidean norm and Frobenius norm operations, respectively. The functions  $\Re(\cdot)$  and  $\Im(\cdot)$  are the real and imaginary components of a complex number, respectively.  $j$  is a complex number. The statistical average is represented by the expectation function  $E(\cdot)$ . The function  $(\cdot)^*$  is the complex conjugate of a complex number. The function  $vec(\cdot)$  is a matrix vectorization function that stacks the column vectors of a matrix on top of each other to form a single column vector.

## 2 Golden Code System Model

In this paper we consider a Golden code system that operates over an  $N_t \times N_r$  wireless MIMO channel where  $N_t = 2$  and  $N_r \geq N_t$  for optimal operation according to [24]. The parameters  $N_t$  and  $N_r$  are the number of transmit and receive antennas in the MIMO configuration, respectively. Golden code works by separating information bitstreams into 4 parallel streams. Each stream has bits packaged into  $\log_2 M$  bit length words and these words are used as symbol indices to select the complex M-QAM symbols from the M-QAM complex signal constellation  $\Omega_M$ . This generates 4 M-QAM complex symbols that are transmitted over a wireless channel by mapping pairs of the M-QAM complex symbols onto the Golden code super symbols. The mapping of M-QAM symbol pairs onto the Golden code super symbols is performed as follows: Let  $x_{11}, x_{12}, x_{21}$  and  $x_{22}$  be the transmitted Golden Code super symbols in which  $x_{11} = \frac{\alpha}{\sqrt{5}}(x_1 + x_2\theta)$  and  $x_{12} = \frac{\alpha}{\sqrt{5}}(x_3 + x_4\theta)$  and  $x_{21} = \frac{\bar{\alpha}}{\sqrt{5}}(x_1 + x_2\bar{\theta})$  and  $x_{22} = j\frac{\bar{\alpha}}{\sqrt{5}}(x_3 + x_4\bar{\theta})$ . The scalar parameters  $\alpha, \bar{\alpha}, \theta$  and  $\bar{\theta}$  are defined as follows:  $\alpha \triangleq 1 + j\bar{\theta}$ ,  $\bar{\alpha} \triangleq 1 + j\theta$ ,  $\theta = \frac{1+\sqrt{5}}{2}$  and  $\bar{\theta} = \frac{1-\sqrt{5}}{2}$ . The complex M-QAM symbols are  $x_1, x_2, x_3, x_4 \in \Omega_M$ , where  $\Omega_M$  is an arbitrary square M-QAM signal constellation. The average M-QAM symbol power is set to 1,  $E(|x_q|^2) = 1, \forall q \in [1:4]$ . These Golden code super symbols are transmitted over the air using the  $N_t$  transmit antennas with the transmit power per antenna fixed to  $\frac{\bar{\gamma}}{N_t}$  where  $\bar{\gamma}$  is the average SNR at each receive antenna.

During wireless transmission, the Golden code transmission vector for timeslot  $i, i \in [1:2]$  is set to  $\mathbf{x}_i = [x_{i1} \ x_{i2}]^T$ . The wireless MIMO channel matrix for timeslot  $i$  is  $\mathbf{H}_i$ , where  $\mathbf{H}_i \in \mathbb{C}^{N_r \times N_t}$  is a fast frequency-flat fading wireless channel which is fully known at the wireless receiver. As a result of the wireless channel being fast fading, this means the wireless channel changes its complex channel gains for every transmission timeslot. The wireless channel fading gain is assumed to be Rayleigh distributed to mimic multipath fading without line of sight (LOS). Each entry of the wireless channel matrix  $\mathbf{H}_i$  varies according to the independent and identically distributed (i.i.d) zero mean complex Gaussian distribution  $\mathcal{CN}(0,1)$ . The received wireless signal vector for timeslot  $i$  is given by (C.1)

$$\mathbf{y}_i = \mathbf{H}_i \mathbf{x}_i + \mathbf{n}_i \tag{C.1}$$

where  $\mathbf{y}_i \in \mathbb{C}^{N_r \times 1}$  is the received signal vector for timeslot  $i$  and  $\mathbf{n}_i \in \mathbb{C}^{N_r \times 1}$  is the noise vector for timeslot  $i$ . Each entry of the zero mean complex Gaussian noise vector,  $\mathbf{n}_i$ , is distributed according to  $\mathbb{CN}\left(0, \frac{N_t}{\gamma}\right)$ . For this work, the detection algorithms used to detect the transmitted M-QAM symbols are the SD-SDS-DNN, SD [14] and SD-SDS [15]. These detection algorithms are evaluated against each other based on BER performance and detection complexity.

### 3 Golden Code SD-SDS Overview

The authors in [15] propose a low complexity detection algorithm called SD-SDS for a generalized Golden code full-rate and full-diversity STBC scheme called multiple complex symbol Golden code (MCS-GC). MCS-GC involves transmitting 2 or more complex M-QAM symbols inside a Golden code super symbol over 2 or more timeslots. The transmission happens over an  $N_t \times N_r$  wireless MIMO channel where  $N_t = 2$  and  $N_r \geq N_t$ . The conventional Golden code system is represented as 2CS-GC in [15] since only 2 complex symbols are transmitted per Golden code super symbol over 2 timeslots. The 2CS-GC system model used by the SD-SDS detection algorithm does not take the form of the one presented in (C.1) in this paper. Instead of a transmission vector of Golden code super symbols as shown in (C.1), the system model uses a transmission vector of complex M-QAM symbols. The Golden code super symbols are just a linear combination of the complex M-QAM symbol pairs and thus the linear combination constants of the Golden code can be factored into the wireless channel matrix and have the transmission vector composed purely of M-QAM symbols. The channel matrix with the Golden code linear combination constants is a modified wireless channel matrix. The equivalence relation that relates the Golden code system model with a transmission vector of Golden code super symbols and the system model with a transmission vector of M-QAM complex symbols is shown in (C.2):

$$\mathbf{H}_i \mathbf{x}_i = \mathbf{H}_i \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix} = \tilde{\mathbf{H}}_i \mathbf{u} \quad (\text{C.2})$$

Where  $\mathbf{u} = [x_1 \quad x_2 \quad x_3 \quad x_4]^T$ ,  $\tilde{\mathbf{H}}_1 = \frac{1}{\sqrt{5}} \mathbf{H}_1 \begin{pmatrix} \alpha & \alpha\theta & 0 & 0 \\ 0 & 0 & \alpha & \alpha\theta \end{pmatrix}$  and  $\tilde{\mathbf{H}}_2 =$

$$\frac{1}{\sqrt{5}} \mathbf{H}_2 \begin{pmatrix} \bar{\alpha} & \bar{\alpha}\bar{\theta} & 0 & 0 \\ 0 & 0 & j\bar{\alpha} & j\bar{\alpha}\bar{\theta} \end{pmatrix}.$$

Wireless channel matrix  $\tilde{\mathbf{H}}_i \in \mathbb{C}^{N_r \times 2N_t}$  is the modified wireless channel matrix, for timeslot  $i$ , that includes the Golden code super symbol linear combination constants based on the equivalence relations in (C.2). The received signal vectors in (C.1), over timeslot  $i$ , are combined using the methodology as shown in [15] to produce (C.3)

$$\mathbf{y} = \tilde{\mathbf{H}}\mathbf{u} + \mathbf{n} \quad (\text{C.3})$$

where  $\tilde{\mathbf{H}} = [\tilde{\mathbf{H}}_1 \ \tilde{\mathbf{H}}_2]^T \in \mathbb{C}^{2N_r \times 2N_t}$ ,  $\mathbf{y} = [\mathbf{y}_1 \ \mathbf{y}_2]^T \in \mathbb{C}^{2N_r \times 1}$ , and  $\mathbf{n} = [\mathbf{n}_1 \ \mathbf{n}_2]^T \in \mathbb{C}^{2N_r \times 1}$ .

To decode the transmitted M-QAM symbols  $x_1, x_2, x_3, x_4$ , SD-SDS is used based on the system model presented in (C.3). QR factorization is first performed on the modified wireless channel matrix  $\tilde{\mathbf{H}}$  such that we get (C.4)

$$\tilde{\mathbf{H}} = \tilde{\mathbf{Q}}\tilde{\mathbf{R}} \quad (\text{C.4})$$

The matrix  $\tilde{\mathbf{Q}} \in \mathbb{C}^{2N_r \times 2N_r}$  is a unitary matrix and matrix  $\tilde{\mathbf{R}} = [\tilde{\mathbf{R}}_1 \ \tilde{\mathbf{R}}_2]^T \in \mathbb{C}^{2N_r \times 2N_t}$  has an upper triangular matrix  $\tilde{\mathbf{R}}_1 \in \mathbb{C}^{2N_t \times 2N_t}$  and also a zero matrix  $\tilde{\mathbf{R}}_2 \in \mathbb{R}^{(2N_r - 2N_t) \times 2N_t}$ . The vector  $\mathbf{z} = \tilde{\mathbf{Q}}^H \mathbf{y} \in \mathbb{C}^{2N_r \times 1}$  is the modified received signal vector over 2-timeslots, which is given by (C.5)

$$\mathbf{z} = \tilde{\mathbf{R}}\mathbf{u} + \tilde{\mathbf{Q}}^H \mathbf{n} \quad (\text{C.5})$$

The low complexity SD-SDS detection algorithm proposed in [15] is summarized as follows:

**SD-SDS Algorithm:**

**Step 1:** Determine the complex M-QAM symbol estimates using QR-decomposition. Estimate  $\hat{x}_q$  where  $q \in [1:4]$ .

$$\begin{aligned} \hat{x}_q &= \frac{\mathbf{z}_q}{\tilde{\mathbf{R}}_{q,q}}, q = 4 \\ &\cdot \\ \hat{x}_q &= \frac{\mathbf{z}_q - \sum_{l=q+1}^4 \tilde{\mathbf{R}}_{q,l} \hat{x}_l}{\tilde{\mathbf{R}}_{q,q}}, \forall q \in [3:1] \end{aligned} \quad (\text{C.6.1}).$$

where  $\mathbf{z}_q$  is the  $q^{th}$  scalar element stored in vector  $\mathbf{z}$  and  $\tilde{\mathbf{R}}_{q,q}$  is the scalar element stored in the  $q^{th}$  row and  $q^{th}$  column of the matrix  $\tilde{\mathbf{R}}$ .

**Step 2:** *Determine the Fixed SD-SDS initial radius*

From [15], the initial radius is calculated based on [20, Eqn (28)].

**Step 3:** *Create the sorted detection subsets*

Using the M-QAM symbol estimates found in (C.6.1), sort in ascending order the M-QAM signal constellation for each estimated symbol based on the Euclidean distance squared metric in (C.6.2). The sorting is done in such a way that the complex symbols in the signal constellation are ordered in ascending order based on which complex symbol is closest to the estimated M-QAM symbols

$$m_q(i) = |\hat{x}_q - u_i|^2, \forall i \in [1: M], u_i \in \Omega_M \quad (\text{C. 6.2}).$$

This implies that for each estimated M-QAM symbol,  $\hat{x}_q$ , we need to find the associated M-QAM symbols, sorted in ascending order, of the  $M - 1$  nearest neighbors from the M-QAM signal constellation. Furthermore, depending on the average SNR, a subset of the sorted M-QAM constellation symbol order is used for detection. The subset lengths ( $L$ ) are stated in [15, Table 2]. The sorting and  $L$ -dimensional subset determination is shown in (C.6.3):

$$S_q = \text{sort}(m_q(i))[1:L] \quad (\text{C. 6.3})$$

where  $S_q, \forall q \in [1: 4]$ , are the  $L$ -dimensional sorted subsets used in the detection of the optimal estimated transmitted symbols  $x_1, x_2, x_3$ , and  $x_4$ .

**Step 4:** *Perform SD-SDS to determine candidates in hypersphere*

The lattice point candidates which are being searched for, using SD-SDS, must lie inside the hypersphere stated in (C.6.4) as per [15]

$$\|\mathbf{z} - \tilde{\mathbf{R}}\mathbf{u}\|_{\text{F}}^2 \leq r^2 \quad (\text{C. 6.4})$$

where  $r$  is the fixed initial radius determined in Step 2. The SDS found in Step 3 are used to search for these lattice point candidates that satisfy (C.6.4).

## 4 Golden Code 2-Timeslot SD-SDS Initial Radius

This section presents two approaches to calculate the SD-SDS initial radius for the 2-timeslot Golden code scheme. The first approach is the fixed initial radius that we will derive for the 2-timeslot scheme and show that the single timeslot initial radius formula used by [15] is valid but was not justified in their paper. The fixed initial radius approach brings the disadvantage of finding many lattice points meeting the constraint stated in (C.6.4), under low average SNR conditions, in situations where the instantaneous SNR is high. This creates high detection complexity at lower average SNR values as it will be shown that at low average SNR values, the initial radius is large. The second approach involves a deep learning model that is used to predict the initial radius. The difference here is that the predicted initial radius depends on the instantaneous channel conditions instead of the average channel conditions. This means for each channel use, the initial radius is adapted to select a minimal number of lattice point candidates and thus reduce complexity. This idea is borrowed from [16] with a modification of the algorithm. Section 4.1 presents the fixed initial radius derivation justifying using a single timeslot initial radius calculation for a 2-timeslot scheme. Section 4.2 presents the adaptive initial radius deep neural network model.

### 4.1 Derivation of Fixed Initial Radius

This section presents the proof of the 2-timeslot Golden code SD-SDS fixed initial radius as dependent only on a single timeslot. Hence, the traditional single timeslot initial radius formulae can be used for the 2-timeslot SD-SDS. We simplify (C.6.4) using (C.5) to get the expression

$$\|\tilde{\mathbf{Q}}^H \mathbf{n}\|_{\text{F}}^2 \leq r^2 \quad (\text{C. 7.1})$$

But we know that  $\tilde{\mathbf{Q}}$  is unitary which implies that  $\tilde{\mathbf{Q}}^H \tilde{\mathbf{Q}} = \mathbf{I}_{2N_r} = \tilde{\mathbf{Q}} \tilde{\mathbf{Q}}^H$  where  $\mathbf{I}_{2N_r} \in \mathbb{R}^{2N_r \times 2N_r}$  is an identity matrix. This also implies that the matrix  $\tilde{\mathbf{Q}}^H$  is unitary. We know from linear algebra that a vector's Frobenius norm is invariant to the multiplication with a unitary matrix [25]. Therefore, we can simplify (C.7.1) to get (C.7.2)

$$\|\tilde{\mathbf{Q}}^H \mathbf{n}\|_F^2 = \|\mathbf{n}\|_F^2 \leq r^2 \quad (\text{C.7.2})$$

*Lemma 1:* We know that the noise vectors  $\mathbf{n}_1$  and  $\mathbf{n}_2$  are  $N_r$ -dimensional and that each entry  $n_w$  is distributed based on the zero-mean complex Gaussian distribution  $\mathcal{CN}(0, \sigma^2)$ . We also know that each noise vector entry is defined as a complex number as show in (C.7.3)

$$n_w = x_w + jy_w \quad (\text{C.7.3})$$

where  $x_w \sim N\left(0, \frac{1}{2}\sigma^2\right)$  and  $y_w \sim N\left(0, \frac{1}{2}\sigma^2\right)$ . Taking the square of a Frobenius norm of a  $2N_r$ -dimensional noise vector  $\mathbf{n}$  yields the following in (C.7.4)

$$\|\mathbf{n}\|_F^2 = \sum_{w=1}^{2N_r} |n_w|^2 = \sum_{w=1}^{2N_r} (x_w^2 + y_w^2) \quad (\text{C.7.4})$$

Using the laws of linear combination of variances, we have (C.7.4) being re-written as (C.7.5)

$$\|\mathbf{n}\|_F^2 = \frac{\sigma^2}{2} \left[ \sum_{w=1}^{2N_r} (N(0,1))^2 + (N(0,1))^2 \right] \quad (\text{C.7.5})$$

But since our sphere decoding algorithm will only search for lattice points using the upper half of the  $2N_r$ -dimensional received signal vector  $\mathbf{z}$ , as per [15], the lower half only contains noise and no signal. Therefore, we only need to consider the upper half of the  $2N_r$ -dimensional noise vector  $\mathbf{n}$ . This means (C.7.6) becomes the relevant expression for our use case

$$\|\mathbf{n}_1\|_F^2 = \frac{\sigma^2}{2} \left[ \sum_{w=1}^{N_r} (N(0,1))^2 + (N(0,1))^2 \right] \sim \frac{\sigma^2}{2} (\varphi_1) \quad (\text{C.7.6})$$

From Lemma 1 it follows that  $\|\mathbf{n}_1\|_F^2$  is a random variable where  $(\varphi_1) \sim \chi^2(2N_r)$  is  $\chi^2$  distributed with  $2N_r$  degrees of freedom. The random variable  $(\varphi_1) \sim \Gamma(\text{shape} = N_r, \text{scale} = 2)$  is also Gamma distributed with a shape of  $N_r$  and a scale of 2. In order to get the sphere decoder initial radius  $r$  we need to set  $\frac{\sigma^2}{2} (\varphi_1) \leq r^2$ , thus  $(\varphi_1) \leq \frac{2r^2}{\sigma^2}$ . Therefore, we can set the probability that the Gamma distributed random variable  $\varphi_1$  is always less than or equal to  $\frac{2r^2}{\sigma^2}$ .  $\text{Prob}\left(\varphi_1 \leq \frac{2r^2}{\sigma^2}\right) = \varepsilon$  where  $\varepsilon \rightarrow 1$ . In our case, we set  $\varepsilon = 0.995$  for 16-QAM and 0.9999 for 64-QAM. This implies that  $\text{Prob}(\varphi_1 \leq P_\varepsilon) = \varepsilon$ . So, we will find the 99.5% or 99.99% percentile value  $P_\varepsilon$  for the Gamma distributed random variable  $\varphi_1$ . Then we derive the sphere decoder initial radius squared as:

$$r^2 = \frac{\sigma^2}{2} P_\varepsilon \quad (\text{C.7.7})$$

But we know that  $\sigma^2 = \frac{N_t}{\bar{\gamma}}$ . This derivation proves that even though Golden code is a 2-timeslot scheme, the fixed initial radius is only dependent on the first timeslot, as shown in (C.7.6). Hence the single timeslot initial radius formulas in literature may be applied in SD-SDS. It also proves that as the average SNR  $\bar{\gamma} \rightarrow 0$  dB then  $r^2$  approaches a large value based on (C.7.7) since  $r^2 \propto \frac{1}{\bar{\gamma}}$ .

## 4.2 Adaptive Initial Radius

In this section we present a Golden Code 2-timeslot deep neural network (DNN) SD-SDS radius predictor by extending the single timeslot DNN SD radius predictor found in [16] to a 2-timeslot DNN. Our DNN SD-SDS radius predictor has inputs from both timeslots 1 and 2. The inputs are stacked into a vector of size  $2N_r(N_t + 1)$  as shown in (C.8)

$$\mathbf{v} = [(\mathbf{y}_1)^T, \text{vec}(\mathbf{H}_1)^T, (\mathbf{y}_2)^T, \text{vec}(\mathbf{H}_2)^T] \quad (\text{C.8})$$

The entries  $(\mathbf{y}_i)^T, i \in [1:2], l \in [1:N_r]$  are the  $N_r$ -dimensional received signal row vectors from (C.1). The entries  $\text{vec}(\mathbf{H}_i)^T, i \in [1:2]$  are the vectorized channel matrix entries from (C.1). We

then convert the complex-valued vector in (C.8) into a real-valued matrix  $\Theta \in \mathbb{R}^{2 \times 4N_r(N_t+1)}$  as shown in (C.9) from [26]

$$\Theta = \begin{pmatrix} \Re(\mathbf{v}) & -\Im(\mathbf{v}) \\ \Im(\mathbf{v}) & \Re(\mathbf{v}) \end{pmatrix} \quad (\text{C. 9})$$

Since this is supervised learning, the offline training of the DNN is done with an output label in the form of the distance squared of the lattice point closest to the upper half of the received signal vector  $\mathbf{z}$ . This translates to the closest lattice radius to the upper half of the received signal vector  $\mathbf{z}$ . The radius is found from the SD-SDS assisted ML detector output distances for each possible combination of the 4 M-QAM symbols. The radius or distance squared for the lattice point with the smallest distance from the received signal vector is computed in (C.10) [16]

$$r_k^2 = \|\mathbf{z} - \tilde{\mathbf{R}}\mathbf{u}^k\|_F^2 \quad (\text{C. 10})$$

where  $\mathbf{u}^k$  is the  $k^{th}$  smallest radius lattice point found inside the hyper-sphere of radius  $r^2$ . The initial radius used for the SD-SDS is based on the derived fixed initial radius in (C.7.7) of this paper. The radii or distances squared for the candidate lattice points are sorted in ascending order as follows:  $r_1^2 < r_2^2 < r_3^2 < \dots < r_K^2$  [16], where  $K$  is a large number of candidates, especially at lower SNR as shown in (C.7.7). The smallest radius squared is loaded into a 1-dimensional vector  $\mathbf{r}=[r_1^2]$  and is used as the output label data for the input training data generated using (C.8) and (C.9).

We only select one radius as the output label because during training we realized that because SD-SDS is a lower complexity detection algorithm relative to SD, sometimes there is only one lattice point that lies within the hypersphere radius. The number of lattice points found inside the hypersphere fluctuates from one-to-many candidates. To collect as many training data points as possible, we select all possible number of lattice points from 1 point to the largest possible number. The input and output training data are normalized or scaled into the interval [0,1]. The scaling for the input feature data  $X$  is carried out using the formula  $X = \left( \frac{X - X_{min}}{X_{max} - X_{min}} \right)$  per input feature.

$X_{min}$  is the smallest feature value over all training SNR values and  $X_{max}$  is the highest feature value over all training SNR values.

The DNN in Table C.1 is trained from 30dB to 2dB SNR range from the highest SNR value to the lowest. The output label data  $Y$  is scaled using the formula  $Y = \left( \frac{Y - Y_{min}}{Y_{max} - Y_{min}} \right)$ .  $Y_{max}$  is the maximum radius squared value of the lattice points and  $Y_{min}$  is the minimum radius squared value over the whole SNR training range. From experimentation, we observe that the output linear activation function used in [16] yields negative radiuses. This motivates us to use a bounded activation function such as the Sigmoid activation function since the radius values cannot be negative and further to that, the radius values do get quite large at low average SNR which negatively affects the training performance of the DNN. The training data is then used to train the neural network in Table C.1 to minimize the mean squared error loss function using the ADAM optimizer [27].

**Table C. 1: DNN Radius Predictor Architecture**

<b>Layer</b>	<b>Parameter Description</b>
Flatten	$1 \times 2 \times 4N_r(N_t + 1)$ input data dimension
(Input-Layer) Dense+LeakyReLU	8 neurons. LeakyReLU slope of 0.3
BatchNormalization	void
Dropout	Probability of 0.5
(Hidden-Layer) Dense+LeakyReLU	8 neurons. LeakyReLU slope of 0.3
BatchNormalization	void
Dropout	Probability of 0.5
(Output-Layer) Dense+Sigmoid	1 neuron
Batch Size=1829, Learning Rate=0.0076, Seed Value=16 and Maximum Epochs=4000	

We notice from observation that the DNN architecture is sensitive to the input Matrix or Vector shape or size. This implies that the DNN architecture in Table C.1 is valid for a  $2 \times 4$  MIMO configuration since our training is based on this MIMO configuration. The baseband modulation schemes used in training the DNN in Table C.1 are the 4-QAM, 8-QAM, 16-QAM and 64-QAM data symbols. For any other MIMO configuration and modulation schemes, a new architecture will need to be selected and trained. During operation, the DNN in Table C.1 will be used as an adaptive initial radius squared predictor based on the normalized instantaneous received signal vectors and wireless channel matrices from timeslot 1 and 2 as per (C.8) and (C.9). The predicted output radius squared from the DNN in Table C.1 is de-normalized back to the initial radius

squared original scale. The formula used for de-normalizing the predicted radius squared is  $r^2 = (Y_{max} - Y_{min}) \times r_{pred}^2 + Y_{min}$  where  $r_{pred}^2$  is the normalized predicted radius squared in the range [0,1] and  $r^2$  is the de-normalized predicted radius squared.

## 5 Golden Code SD-SDS Deep Learning Algorithms

This section is dedicated to exhibiting the algorithms developed to lower complexity in Golden code SD-SDS detection. Section 5.1 ventilates the SD-SDS-Radius-DNN algorithm that aims to lower the detection complexity of SD-SDS at mid-to-low SNR. This algorithm uses the adaptive initial radius DNN predictor, in Table C.1, to predict the SD-SDS initial radius based on the instantaneous channel conditions as per [16]. Section 5.2 exhibits the novel SD-SDS-DNN algorithm that executes the SD-SDS-Radius-DNN algorithm under unfavorable instantaneous channel or noise conditions and the QR decomposition sub-optimal detector is preferred in favorable instantaneous channel and noise conditions. The SD-SDS-DNN algorithm lowers the detection complexity of the SD-SDS at low SNR by preferably running the low complexity QR decomposition sub-optimal detector under favorable instantaneous channel and noise conditions.

### 5.1 SD-SDS-Radius-DNN Algorithm

#### SD-SDS-Radius-DNN algorithm:

**Input:**  $\mathbf{y}_1, \mathbf{y}_2, \mathbf{H}_1, \mathbf{H}_2, \mathbf{z}, \tilde{\mathbf{R}}, \Omega_M, \bar{\gamma}$

**Output:**  $\hat{t}_1, \hat{t}_2, \hat{t}_3, \hat{t}_4$

1.  $S_q = SDS(\mathbf{z}, \tilde{\mathbf{R}}, \Omega_M) \forall q \in [1:4]$
2. *If*  $\bar{\gamma} > \gamma_t$  *Then*
  - a.  $r^2 = \frac{\sigma^2}{2} P_\varepsilon$
  - b.  $c_{exit} = 0$
3. *Else*
  - a. Stack  $\mathbf{y}_1, \mathbf{y}_2, \mathbf{H}_1, \mathbf{H}_2$  entries as shown in (C.8) and (C.9) to obtain  $\Theta$
  - b. Feed  $\Theta$  to the trained DNN in Table C.1 to get the radius squared prediction  $r^2 = DNN_{radius}(\Theta)$
  - c.  $c_{exit} = 4$
4.  $c = 0$
5. Implement sphere decoding with initial radius  $r^2$
6. *If*  $D_{sp}(\mathbf{z}, \tilde{\mathbf{R}}, S_q, r^2) \neq null$  *OR*  $c = c_{exit}$  *Then*
  - a. *GO TO Step 8*
7. *Else*
  - a.  $c = c + 1$

- b.  $r^2 = r^2 * 2$
- c. *GO TO Step 5*
- 8. *If  $D_{sp}(\mathbf{z}, \tilde{\mathbf{R}}, S_q, r^2) \neq \text{null}$  Then*
  - a.  $\hat{l}_1, \hat{l}_2, \hat{l}_3, \hat{l}_4 = D_{sp}(\mathbf{z}, \tilde{\mathbf{R}}, S_q, r^2)$
- 9. *Else*
  - a.  $\hat{l}_1, \hat{l}_2, \hat{l}_3, \hat{l}_4 = D_{sub}(\mathbf{y}_1, \mathbf{y}_2, \mathbf{H}_1, \mathbf{H}_2, \bar{\gamma})$
- 10. *End*

where  $D_{sp}(\cdot)$  is the SD algorithm as implemented in [15],  $D_{sub}(\cdot)$  is the sub-optimal detector of M-QAM symbols when the SD algorithm finds no lattice points candidates.  $\gamma_t$  is the average SNR threshold below which the adaptive initial radius DNN algorithm is activated.  $\gamma_t$  is defined as 11 dB for 16-QAM and 19 dB for 64-QAM. The SD-SDS algorithm has low detection complexity at high SNR and higher detection complexity at lower SNR values as stated in [15]. The values 11 dB and 19 dB are found via experimentation after observing that the SD-SDS-Radius-DNN algorithm effectively reduces complexity below a specific average SNR threshold. The 16-QAM case was found to be effective from 11 dB downwards and for 64-QAM it was found to lower detection complexity from 19 dB downwards. In this paper, we are targeting the high detection complexity at lower SNR values for SD-SDS. Therefore, the thresholds basically determine when the adaptive initial radius predictor must take effect and reduce detection complexity.  $SDS(\cdot)$  is the sorted detection subset algorithm as detailed in Step 3 of Section 3.  $\hat{l}_1, \hat{l}_2, \hat{l}_3, \hat{l}_4$  are the decoded M-QAM symbol indices.

The SD-SDS-Radius-DNN algorithm is a modified version of the algorithm in [16]. The SD-SDS-Radius-DNN algorithm is modified to suit the SD-SDS context of having only 1 predicted radius squared output. This means that if no lattice point candidates are found within the predicted radius squared of the hypersphere, we need a way to increase the hypersphere radius and perform the sphere-decoding again. We are inspired by the idea in [17] of iteratively dividing the hypersphere radius by 2 and feeding the updated radius to a DNN to predict the number of lattice points inside a hypersphere. In our case, we do the opposite; if we do not find lattice point candidates inside the hypersphere, we increase the radius squared by multiplying it by a factor of 2 until we reach a pre-determined iteration limit of 5 set as  $c_{exit} = 4$ . The  $c_{exit}$  limit is set via a heuristic method by considering that a high iteration limit leads to higher detection complexity or latency. A lower iteration limit leads to sub-optimal BER performance

as the sub-optimal detector will be relied on more often. The algorithm also has the DNN prediction of the initial radius squared only done below a certain average SNR threshold since SD-SDS has a high detection complexity at lower SNR values whilst at higher SNR values, it has low detection complexity. When the SD-SDS-Radius-DNN algorithm fails to find lattice point candidates inside the hypersphere of a pre-determined radius squared, the LMMSE estimator  $\hat{\mathbf{x}}_i = (\mathbf{H}_i^H \mathbf{H}_i + \sigma^2 \mathbf{I}_{N_t})^{-1} \mathbf{H}_i^H \mathbf{y}_i$   $i \in [1:2]$  from [16] is used as a sub-optimal detector for Golden code symbol estimates. The vector  $\hat{\mathbf{x}}_i$  is the sub-optimally estimated transmitted Golden code symbol vector for timeslot  $i$ . The 4 square M-QAM symbols conveyed by these estimated Golden code symbols are then found using expression (C.11.1) to (C.11.2) which are adapted from [14]

$$\hat{x}_{2k-1} = \frac{\sqrt{5}}{(\theta - \bar{\theta})} \left\{ \theta \frac{\hat{x}_2[k-1]}{\bar{\alpha}} - \bar{\theta} \frac{\hat{x}_1[k-1]}{\alpha} \right\} \quad (\text{C. 11.1})$$

$$\hat{x}_{2k} = \frac{\sqrt{5}}{(\theta - \bar{\theta})} \left\{ \frac{\hat{x}_1[k-1]}{\alpha} - \frac{\hat{x}_2[k-1]}{\bar{\alpha}} \right\} \quad (\text{C. 11.2})$$

where  $k \in [1:2]$ .

Therefore, we can search for the symbol indices that minimize the following Euclidean distances squared based on (C.12) for each estimated  $\hat{x}_q$  square M-QAM symbol.

$$\hat{i}_q = \underset{u_i \in \Omega_M}{\text{argmin}} \left( |\hat{x}_q - u_i|^2 \right), \forall i \in [1:M], u_i \in \Omega_M \quad (\text{C. 12})$$

## 5.2 SD-SDS-DNN Novel Algorithm

The SD-SDS-DNN algorithm uses the instantaneous channel conditions and received signal vectors to predict, using a DNN, whether the sub-optimal QR decomposition M-QAM symbol estimates, found in Step 1 of the SD-SDS algorithm in Section 3, are good enough to be used as the actual transmitted symbols without performing the SD-SDS-Radius-DNN based search. The output of the DNN is a probability value in the range [0,1] that is used to determine whether the channel conditions and instantaneous noise values in the received signal vectors are good

enough to directly output the transmitted M-QAM symbols from the QR decomposition sub-optimal detector. If the probability is greater than a specific threshold, then the QR decomposition sub-optimal detector output is taken as the transmitted symbols. If it is less, then the more complex SD-SDS-Radius-DNN based search is performed.

Let us define the proposed DNN which predicts the channel condition state probability based on the instantaneous channel conditions and noise statistics. We define the DNN as

$$p = \Phi(\boldsymbol{\pi}, \boldsymbol{\tau}) \quad (\text{C.13})$$

where  $p \in [0,1]$  is the probability that the channel conditions are sufficient to use the QR decomposition sub-optimal detector estimated M-QAM symbols as the transmitted M-QAM symbols. The function  $\Phi(\cdot)$  is the DNN channel condition state predictor which takes the input vector  $\boldsymbol{\pi} \triangleq [\hat{\mathbf{z}}^R, \hat{\mathbf{z}}^I, \tilde{\mathbf{r}}_1^R, \tilde{\mathbf{r}}_1^I] \in \mathbb{R}^{28}$  which is a combination of the modified received signal vector in (C.5) and the upper triangular matrix  $\tilde{\mathbf{R}}_1$ . The entries of the vector  $\boldsymbol{\pi}$  are real valued as the DNN function approximator can only take real numbers. We define  $\hat{\mathbf{z}}^R \triangleq (\Re(\mathbf{z}[0:N_r]))^T$ ,  $\hat{\mathbf{z}}^I \triangleq (\Im(\mathbf{z}[0:N_r]))^T$ ,  $\tilde{\mathbf{r}}_1^R \triangleq \text{vec}(\Re(\tilde{\mathbf{R}}_1))^T$  and  $\tilde{\mathbf{r}}_1^I \triangleq \text{vec}(\Im(\tilde{\mathbf{R}}_1))^T$ . The notation  $\mathbf{z}[0:N_r)$  means that we take the first  $N_r$  elements of the vector  $\mathbf{z}$  since the last  $N_r$  elements are just noise without any signal. With regards to the upper triangular matrix  $\tilde{\mathbf{R}}_1$ , we only take the non-zero real valued elements of the matrix entries. The DNN input vector  $\boldsymbol{\pi}$  is a 28-dimensional vector because the DNN architecture in Table C.2 is trained for a fixed MIMO configuration of  $N_t = 2$  and  $N_r = 4$ . For any other MIMO configuration the DNN architecture in Table C.2 will change and will need to be re-trained. The vector  $\boldsymbol{\tau}$  is a vector of all hyperparameters that need to be tuned during the training phase of the DNN.

The DNN function approximator  $\Phi(\cdot)$  has an architecture shown in Table C.2.

**Table C. 2: DNN Channel Condition State Predictor Architecture**

Layer	Parameter Description
Input (28,)	28 dimensional input vector $\boldsymbol{\pi}$
(Input Layer) Dense+LeakyReLU	$N_i$ neurons. LeakyReLU slope of 0.3

Layer	Parameter Description
BatchNormalization	void
Dropout	Probability of 0.5
(Hidden Layer) Dense+LeakyReLU	$N_h$ neurons. LeakyReLU slope of 0.3
BatchNormalization	void
Dropout	Probability of 0.5
(Output Layer) Dense+Sigmoid	1 neuron
Batch Size=1800, Learning Rate=0.00009, Seed Value=16 and Maximum Epochs=2000	

where  $N_i$  is the number of input layer neurons and  $N_h$  is the number of hidden layer neurons. For the architecture in Table C.2 to be useful, we need to train the DNN architecture with appropriate training samples and test the DNN before deploying it. The next section attends to this.

### 5.2.1 DNN Training and Testing Phase

The objective of this phase is to tune the weights and biases of the DNN function which are found in the vector  $\boldsymbol{\tau}$ . The DNN function in Table C.2 is trained using approximately 10 000 samples of data over an average SNR range of [0,21] dB for 16-QAM modulation and [10,28] dB for 64-QAM modulation. The DNN is trained from the highest average SNR value to the lowest average SNR value. The way the training data is collected is based on the following pseudocode:

#### **Training Pseudocode:**

###Start Comment ###

### X is the input sample array and Y is the output label sample array

###end comment###

Step 1: Initialize dynamic arrays  $X = [], Y = []$ .

Step 2: counter = 0, sampleSize = 10000

Step 3: If counter < sampleSize Then

a): If  $\hat{x}_1 = x_1$  AND  $\hat{x}_2 = x_2$  AND  $\hat{x}_3 = x_3$  AND  $\hat{x}_4 = x_4$  Then

i)  $X[\text{counter}] = \boldsymbol{\pi}$

ii)  $Y[\text{counter}] = 1$

b) Else

i)  $X[\text{counter}] = \boldsymbol{\pi}$

ii)  $Y[\text{counter}] = 0$

counter = counter + 1

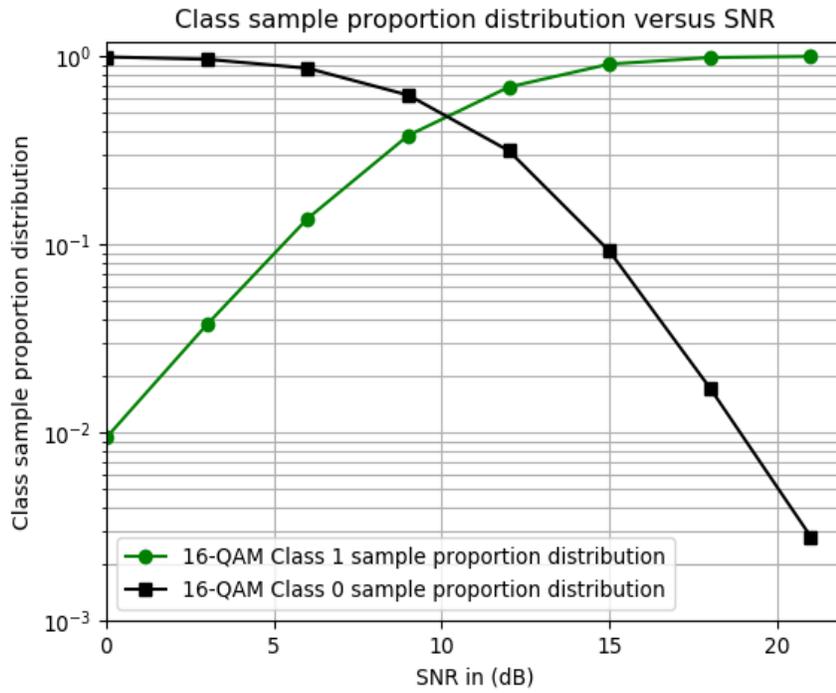
GOTO Step 3

*Step 4: Else*  
*GOTO Step 5*

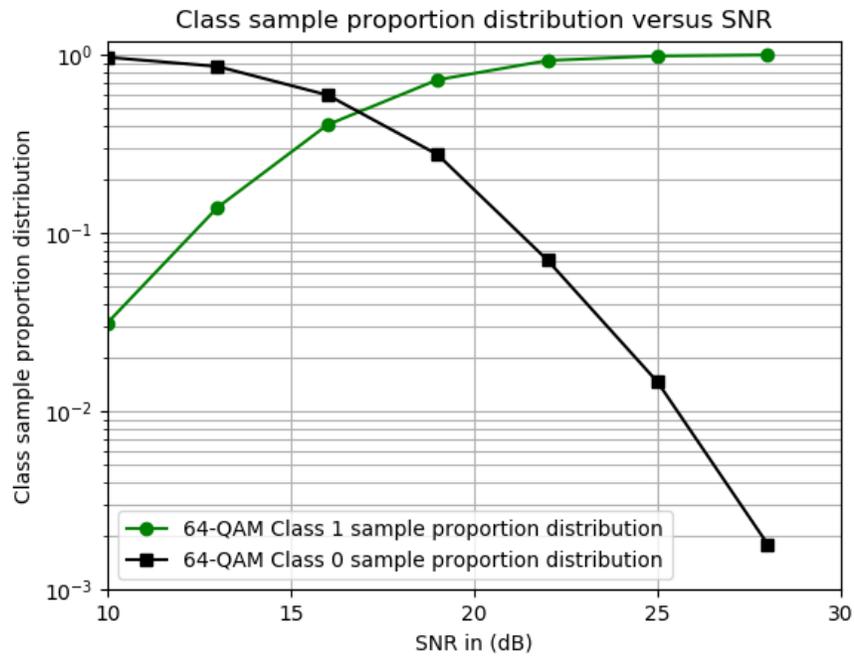
*Step 5: End*

The idea is that when the instantaneous channel conditions are very good, then the condition in Step 3a) can be easily met and the training output label value is set to 1. When the instantaneous channel conditions are not favorable then the condition in 3a) is most likely not met and the output label value is set to 0. The reasoning is that when the instantaneous fading channel conditions are good and instantaneous noise power is low, the QR decomposition M-QAM symbol estimates  $\hat{x}_q, \forall q \in [1,4]$  will be a good enough estimate for the actual transmitted M-QAM symbols  $x_q, \forall q \in [1,4]$ . After collecting the input and output training data, since this is a supervised learning problem, we trained the DNN and realized that the validation accuracy of the DNN under test conditions was in excess of 99% for high and low SNR values. In the mid-SNR range, the validation accuracy went as low as 36%.

Upon inspection of the training data, we realized that the collected training samples were skewed. The output label distribution was not evenly distributed between the 0 state and 1 state for both high and low average SNR values. Figs. C.1 and C.2 illuminate the distribution of the output label states for 16-QAM and 64-QAM, respectively.



**Fig. C. 1 16-QAM skewed output label distribution for the training data.**



**Fig. C. 2 64-QAM skewed output label distribution for the training data.**

This explained why the DNN was underperforming in the mid-SNR range whilst performing in the high and low SNR region. This is because the DNN became lazy to learn and decided to memorize the output state and retain the output predicted state on one state depending on whether it is the low SNR or high SNR region. In the high SNR region, the DNN will constantly output a predicted state of 1 because over 99% of the output labels have values of 1. The objective is to maximize the validation accuracy and the DNN can achieve this by just outputting a predicted state of 1 for the high SNR case. The same logic applies for the low SNR case.

To force the DNN to learn during training, we decided to perform over-sampling of the minority state/class and under-sampling of the majority state/class for the full SNR range. The over-sampling is performed using the synthetic minority over-sampling technique (SMOTE) [29] and the under-sampling is performed using randomized under-sampling of the majority class [29]. The SMOTE works by randomly selecting a minority class/state feature sample in the training data and then using the  $K$ -nearest neighbor (KNN) algorithm to select the  $K$  nearest neighbors to that selected feature sample. It then randomly selects one nearest neighbor from the selected  $K$  neighbors and randomly creates a new synthetic feature sample point on the line joining the chosen nearest neighbor feature sample and the originally selected feature sample on the  $\mathbb{R}^{28}$  dimensional cartesian plane. The process is repeated until a desired sample size of the minority class is achieved. The under-sampling is performed by randomly selecting a feature sample in the majority class/state and then deleting it from the training samples. This is repeated until a desired ratio between the majority class and minority class is achieved. In our case, we performed this until the majority class was approximately 60% of the training sample size and the minority class 40% for the full average SNR range for both 16-QAM and 64-QAM training data.

We then went ahead to train the DNN architecture in Table C.2 using this modified training data with 25% of the training samples being used for testing purposes and 75% for training. The input feature training data in array  $X$  was scaled into the range [0,1] using the methodology explained in Section 4.2. The loss function selected for the optimization of the DNN hyperparameters was the binary cross entropy loss function with the validation accuracy used as the metric to measure performance. The ADAM optimizer [27] was used to perform the optimization of the DNN weights and biases in the vector  $\tau$  by comparing the output of the DNN

to the target values in array  $Y$ . During the testing phase, the DNN is fed multiple test vectors  $\boldsymbol{\pi}$  from the test samples and the output of the DNN predicts a probability value in the range  $[0,1]$ . Table C.3 shows the probability thresholds  $p_t$  and number of neurons used in the architecture for the case of 16-QAM and 64-QAM. The probability thresholds are used to determine the channel condition state. If the predicted probability exceeds a given probability threshold  $p_t$ , then the channel condition state is 1. If the predicted probability is less than or equal to the probability threshold  $p_t$ , then the channel condition state is 0.

**Table C. 3: 16-QAM and 64-QAM DNN Channel Condition State Predictor Parameters**

	<b>16-QAM Parameters</b>		<b>64-QAM Parameters</b>	
Probability Thresholds	$\bar{\gamma} \leq 9 \text{ dB}$	$\bar{\gamma} > 9 \text{ dB}$	$\bar{\gamma} \leq 16 \text{ dB}$	$\bar{\gamma} > 16 \text{ dB}$
	$p_t = 0.5$	$p_t = 0.9$	$p_t = 0.5$	$p_t = 0.9$
Number of Neurons	$N_i = 160$	$N_h = 160$	$N_i = 64$	$N_h = 64$

### 5.2.2 SD-SDS-DNN Algorithm Explained

The DNN function approximator in Table C.2 is used to predict when the QR decomposition M-QAM symbol estimates can be used as appropriate estimates for the transmitted M-QAM symbols. The SD-SDS-DNN algorithm combines this DNN function approximator in Table C.2, the SD-SDS algorithm developed in [15] and the SD-SDS-Radius-DNN algorithm. The SD-SDS-DNN algorithm is presented as follows:

**SD-SDS-DNN algorithm:**

**Input:**  $\mathbf{y}_1, \mathbf{y}_2, \mathbf{H}_1, \mathbf{H}_2, \mathbf{z}, \tilde{\mathbf{R}}, \tilde{\mathbf{R}}_1, \Omega_M, \bar{\gamma}$

**Output:**  $\hat{i}_1, \hat{i}_2, \hat{i}_3, \hat{i}_4$

1.  $\hat{i}_1, \hat{i}_2, \hat{i}_3, \hat{i}_4 = \text{getQRDecompositionSymbolEstimates}()$  from Step 1 of SD-SDS algorithm in Section III

2. *If*  $\bar{\gamma} \leq \zeta_t$  *Then*

a.  $\hat{\mathbf{z}}^R = (\Re(\mathbf{z}[0:N_r]))^T$ ,  $\hat{\mathbf{z}}^I = (\Im(\mathbf{z}[0:N_r]))^T$ ,  $\tilde{\mathbf{r}}_1^R = \text{vec}(\Re(\tilde{\mathbf{R}}_1))^T$  and  $\tilde{\mathbf{r}}_1^I = \text{vec}(\Im(\tilde{\mathbf{R}}_1))^T$

b.  $\boldsymbol{\pi} = [\hat{\mathbf{z}}^R, \hat{\mathbf{z}}^I, \tilde{\mathbf{r}}_1^R, \tilde{\mathbf{r}}_1^I]$

c.  $p = \Phi(\boldsymbol{\pi}, \boldsymbol{\tau})$

d. *If*  $p > p_t$  *Then*

i. *GOTO* Step 4

e. *Else*

- i.  $\hat{i}_1, \hat{i}_2, \hat{i}_3, \hat{i}_4 = \text{run SD-SDS-Radius-DNN Algorithm}$
- 3. *Else*
  - a.  $\hat{i}_1, \hat{i}_2, \hat{i}_3, \hat{i}_4 = \text{run SD-SDS Algorithm}$
- 4. *End*

From the SD-SDS-DNN algorithm we can see that when the average SNR  $\bar{\gamma}$  exceeds the threshold  $\zeta_t$  the SD-SDS algorithm from [15] is executed. This is because at high SNR values the detection complexity of the SD-SDS algorithm is very low. The objective of this paper is to reduce the detection complexity of the SD-SDS algorithm at low SNR values. At lower SNR values,  $\bar{\gamma} \leq \zeta_t$ , the SD-SDS-DNN algorithm executes the DNN channel predictor in Table C.2 that is used to select between the very low complexity sub-optimal QR decomposition detector and the execution of the low complexity near-optimal SD-SDS-Radius-DNN detector. The average SNR thresholds for 16-QAM and 64-QAM modulation are  $\zeta_t = 6$  dB and  $\zeta_t = 19$  dB, respectively. The thresholds  $p_t$  and  $\zeta_t$  are found using a heuristic method.

From the SD-SDS-DNN algorithm we observe that when the predicted probability from the DNN exceeds the probability threshold  $p_t$ , as shown in Table C.3, then the SD-SDS-Radius-DNN algorithm does not get executed at all. The estimated transmitted M-QAM symbols come directly from the very low complexity QR decomposition sub-optimal detector. If the predicted probability is less than or equal to the probability threshold  $p_t$ , then the SD-SDS-Radius-DNN algorithm gets executed as the QR decomposition sub-optimal detector output is deemed unreliable by the DNN predictor.

## 6 Complexity Analysis of Proposed Algorithms

In this section, we will deal with the detection complexity analysis of the three different Golden code detection algorithms discussed here and in literature. The detection complexity is defined in various ways using different metrics. We will extend the detection complexity metric of evaluating complexity using the number of complex-valued operations [30] performed by a detection algorithm. Since some of the algorithms rely on deep neural networks, which only process real values, we will only consider complexity analysis of real-valued floating-point operations (FLOPS) [14]. The real-valued binary operators of interest are the multiplication, addition, subtraction, and division as per [14]. We also neglect the sub-optimal detector's detection complexity since it is only executed 0.5% of the time for 16-QAM and 0.01% for 64-

QAM Monte-Carlo simulations. Its contribution to the average detection complexity is marginal. The DNN complexity analysis of the offline training and data collection is ignored because offline training is done only once [16]. We are only performing the DNN complexity analysis for the online decoding process.

In [16], the detection complexity metric used is the decoding time and order of execution, whilst [17] uses the number of lattice points inside the hypersphere and the average processing time for the decoding process. Our complexity analysis is based on the number of floating-point operations; hence we cannot use any of the complexity results in [16] and [17] as a benchmark against our detection complexity. Our complexity analysis is relative between the Alamouti linear ML, Golden code SD, SD-SDS, and SD-SDS-DNN algorithms for the  $2 \times 4$  MIMO configuration. In [15], they use the number of Euclidean distance calculations as the metric for the complexity analysis of SD-SDS. This again is different from our metric and makes the results in [15] not comparable to ours.

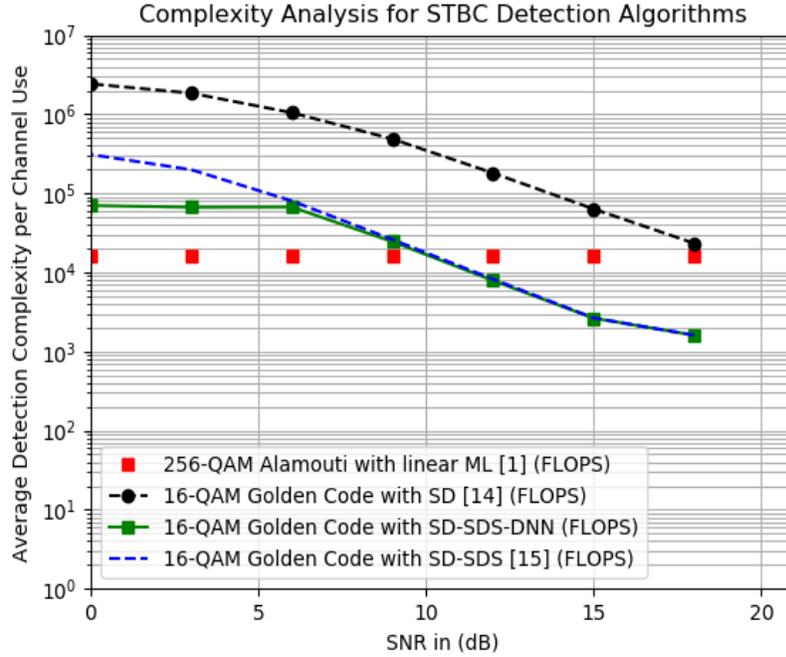
The detection complexity of the three Golden code detection algorithms, SD, SD-SDS, and SD-SDS-DNN is determined using simulations. The results are exhibited in Figs. C.3 and C.4. The detection complexity of the Alamouti STBC linear ML detector is also exhibited in Figs. C.3 and C.4. The Golden code detection algorithms are evaluated in a fast-fading wireless environment; however, it is interesting to see its performance against the Alamouti STBC in a block-fading wireless channel. The reason for this is that we aim to achieve lower detection complexity relative to the low complexity Alamouti linear ML detector which performs optimally in block-fading wireless channels [4]. The Alamouti STBC scheme's modulation order is selected as 256-QAM so that the spectral efficiency of the scheme matches that of the 16-QAM Golden code STBC scheme which is 8 bits/s/Hz. The Alamouti STBC scheme's modulation order is also selected as 4096-QAM so that the spectral efficiency of the scheme matches that of the 64-QAM Golden code STBC scheme which is 12 bits/s/Hz. This makes the comparison fair between the two schemes as we want to see the detection complexity of the two competing STBC schemes for the same achieved spectral efficiency.

As can be seen in Fig. C.3, the proposed SD-SDS-DNN algorithm outperforms the SD-SDS algorithm developed in [15] at low SNR. We observe that the detection complexity is identical between 6 dB and 18 dB because the SD-SDS algorithm is executed when the average SNR exceeds the 6 dB threshold as per the SD-SDS-DNN algorithm. This is because the SD-SDS algorithm exhibits low detection complexity at high SNR values. The low complexity SD-SDS-Radius-DNN algorithm, that runs within the SD-SDS-DNN algorithm, only has an opportunity to be executed when the average SNR is equal to or less than 6 dB. From 6 dB and below, the SD-SDS-DNN algorithm exhibits lower detection complexity relative to the SD-SDS algorithm. This is because the low complexity QR decomposition detector or the SD-SDS-Radius-DNN algorithm are being selected as M-QAM symbol detectors depending on instantaneous channel conditions and noise. In Table C.4, we get the insight of the percentage distribution of how many times the predicted transmitted symbols were determined by the sub-optimal QR decomposition method, SD-SDS decoder or the SD-SDS-Radius-DNN decoder, for 16-QAM modulation.

**Table C. 4: Estimated Transmitted Symbols Decoder Execution Percentage Distribution for 16-QAM**

<b>SNR (dB)</b>	0	3	6	9	12	15	18
SD-SDS-Radius-DNN Decoder (%)	31.4	31.6	36.7	0	0	0	0
Sub-optimal QR Decomposition Decoder (%)	68.6	68.4	63.3	0	0	0	0
SD-SDS Decoder (%)	0	0	0	100	100	100	100

It is clear from Table C.4 that the SD-SDS algorithm is solely used at high SNR for its low detection complexity. For 6 dB SNR and below, the QR decomposition sub-optimal detector is used in most cases to predict the transmitted symbols compared to the SD-SDS-Radius-DNN decoder. Hence the 75% reduction in detection complexity at low SNR relative to the SD-SDS decoder, as shown in Fig. C.3, is due to the mix in the low complexity detection of the QR decomposition sub-optimal detector and the SD-SDS-Radius-DNN decoder.



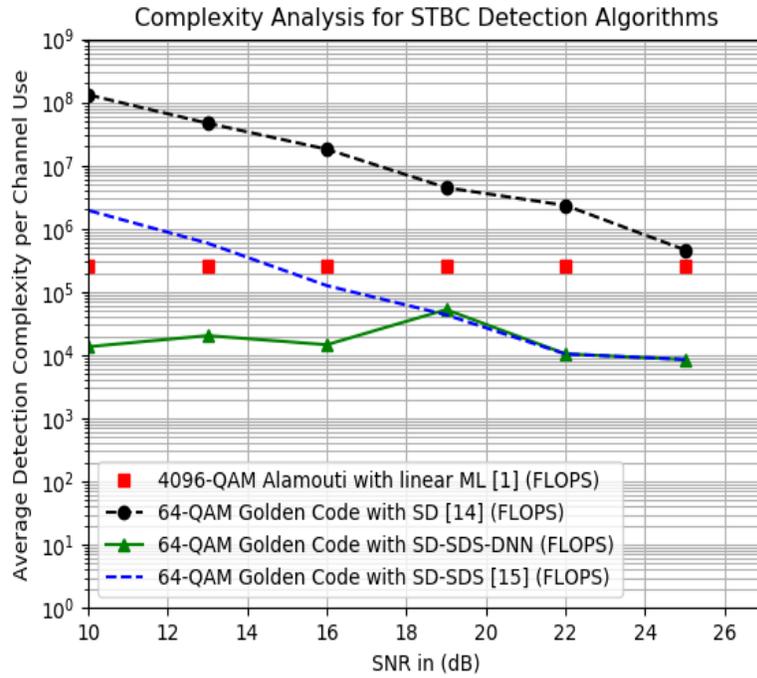
**Fig. C. 3 Complexity analysis of  $2 \times 4$  MIMO 16-QAM Golden Code detection algorithms versus 256-QAM Alamouti detection algorithm**

In Fig. C.3, we also notice that the SD-SDS-DNN algorithm has a detection complexity that is 4 times greater than that of the Alamouti STBC linear ML detector, at low SNR. At high SNR, the SD-SDS-DNN algorithm has 90% lower detection complexity relative to the Alamouti STBC linear ML detector. This shows that Golden code has a future in practical MIMO applications since the detection complexity has been reduced such that it is comparable to that of the Alamouti linear ML detector. With regards to the traditional SD algorithm, it is shown in Fig. C.3 that its detection complexity is the highest amongst the STBC detection algorithms discussed in this paper.

In Fig. C.4, the proposed SD-SDS-DNN algorithm outperforms the SD-SDS algorithm developed in [15] at low SNR. We observe that the complexity is identical between 19 dB and 25 dB because the SD-SDS algorithm is executed as a detector of choice above the 19 dB average SNR threshold as per the SD-SDS-DNN algorithm. Below 19 dB, the instantaneous channel and noise conditions are used to select between the very low complexity sub-optimal QR decomposition detector and the low complexity SD-SDS-Radius-DNN detector. When the instantaneous channel and noise conditions are very good, the DNN channel condition predictor

in Table C.2 selects the very low complexity QR decomposition detector as a detector of choice. When the instantaneous channel conditions are unfavorable, the more complex SD-SDS-Radius-DNN detector is executable as it produces reliable symbol estimates even when the instantaneous channel and noise conditions are unfavorable.

In Table C.5, we exhibit the percentage distribution of how many times the predicted transmitted symbols were determined by the sub-optimal QR decomposition method, SD-SDS decoder or the SD-SDS-Radius-DNN decoder, for 64-QAM modulation.



**Fig. C. 4 Complexity analysis of  $2 \times 4$  MIMO 64-QAM based Golden code detection algorithms versus 4096-QAM Alamouti detection algorithm.**

**Table C. 5: Estimated Transmitted Symbols Decoder Execution Percentage Distribution for 64-QAM**

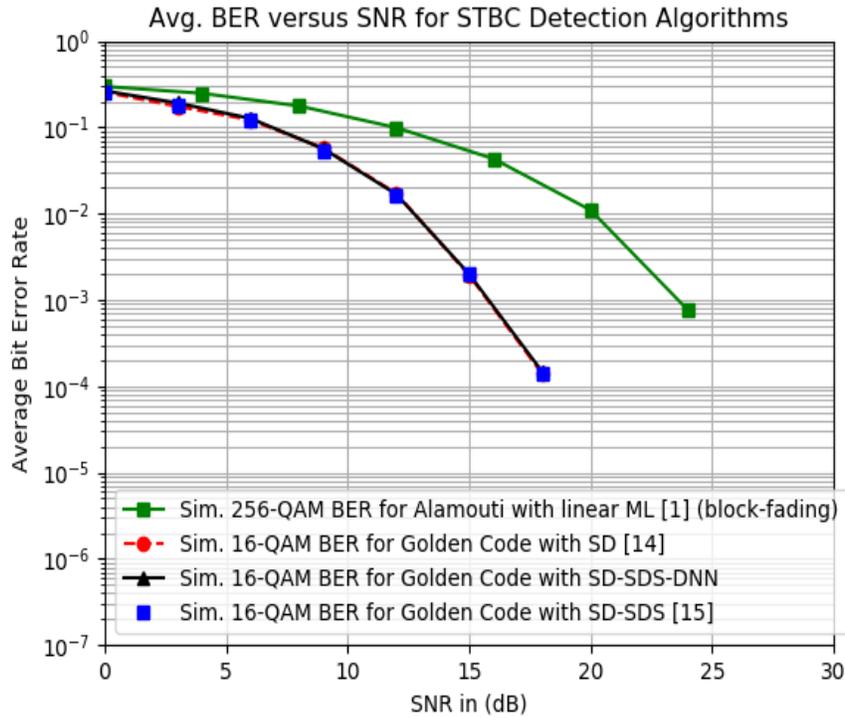
SNR (dB)	10	13	16	19	22	25
SD-SDS-Radius-DNN Decoder (%)	0.5	1.6	1.1	99.8	0	0
Sub-optimal QR Decomposition Decoder (%)	99.5	98.4	98.9	0.2	0	0
SD-SDS Decoder (%)	0	0	0	0	100	100

It is clear from Table C.5 that the SD-SDS algorithm is solely used at high SNR for its low detection complexity. For 16 dB SNR and below, the QR decomposition sub-optimal detector is used at least 98% of the time to predict the transmitted symbols compared to the less than 2% utilization of the SD-SDS-Radius-DNN decoder. Hence the 99% lower detection complexity, at low SNR, relative to the SD-SDS decoder, as shown in Fig. C.4, is largely due to the low complexity detection of the QR decomposition sub-optimal detector.

From Fig. C.4, we observe that the proposed SD-SDS-DNN algorithm outperforms the Alamouti linear ML detector by exhibiting a detection complexity that is 90% lower for the greater part of the SNR range. We also observe that the traditional SD algorithm is the most computationally complex detection algorithm relative to the STBC detection algorithms discussed in this paper.

## 7 Simulation Results and Discussions

The Monte-Carlo wireless simulation environment was setup as a  $2 \times 4$  MIMO, where  $N_t = 2$  and  $N_r = 4$ , with a wireless channel with Rayleigh frequency-flat fast fading in which the channel gain changes per transmission timeslot. The wireless channel transmit-and-receive antennas are sufficiently spaced such that the wireless channels are de-correlated. The information symbol modulation order used in the simulation was 16-QAM and 64-QAM. The average power constraint for the 16-QAM and 64-QAM symbols was set to 1. The SD fixed initial radius probability was set to  $\varepsilon=0.995$  for 16-QAM and  $\varepsilon=0.9999$  for 64-QAM. The Monte-Carlo simulation determined the BER performance of the three detection algorithms SD, SD-SDS and SD-SDS-DNN for 16-QAM and 64-QAM. We also simulated the BER performance of the 256-QAM Alamouti STBC scheme within a block-fading wireless channel. We chose the 256-QAM constellation for the Alamouti scheme so that the spectral efficiency of the Alamouti STBC and 16-QAM Golden code schemes were identical. This allowed us to compare the BER performance of Alamouti against that of the Golden code STBC. The 64-QAM Golden code STBC produced the same spectral efficiency as the 4096-QAM Alamouti scheme.

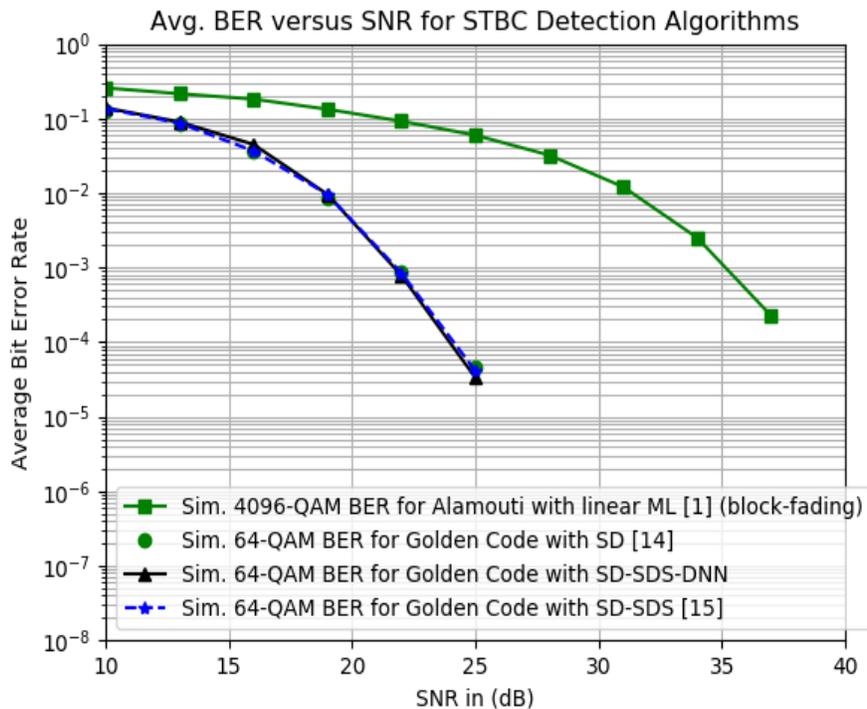


**Fig. C. 5 BER for  $2 \times 4$  MIMO 16-QAM Golden Code Sphere Decoding Detection Algorithms versus 256-QAM Alamouti Detection**

If we look at Fig. C.5, we see that the proposed SD-SDS-DNN algorithm achieves the same BER performance as SD and SD-SDS algorithms from literature for 16-QAM. This means that the detection algorithm achieves the objective of lowering the detection complexity of SD-SDS, at low SNR, without compromising the BER performance. The Golden code STBC scheme has an 8 dB signal power gain over the Alamouti STBC scheme at a BER of  $10^{-3}$  for a spectral efficiency of 8 bits/s/Hz. This implies that the Golden code STBC can achieve the same spectral efficiency as the Alamouti STBC scheme but at a much higher link reliability and comparable detection complexity with the linear ML detector as shown in Fig. C.3.

In Fig. C.6, we see that the proposed SD-SDS-DNN algorithm achieves the same BER performance as SD and SD-SDS algorithms from literature for 64-QAM. This means that the detection algorithm achieves the objective of lowering the detection complexity of SD-SDS, at low SNR, without compromising the BER performance. The Golden code STBC scheme has a 13 dB signal power gain over the Alamouti STBC scheme at a BER of  $10^{-3}$  for a spectral

efficiency of 12 bits/s/Hz. This implies that the Golden code STBC can achieve the same spectral efficiency as the Alamouti STBC scheme but at a much higher link reliability and 90% lower detection complexity relative to the linear ML detector as shown in Fig. C.4.



**Fig. C. 6 BER simulation results for  $2 \times 4$  MIMO 64-QAM Golden Code Sphere Decoding Detection Algorithms versus 4096-QAM Alamouti Detection**

## 8 Conclusion and Future Work

The SD-SDS-DNN algorithm was developed in our research to lower detection complexity of SD-SDS, at low SNR, whilst maintaining the BER performance. The SD-SDS-DNN algorithm is shown to reduce the detection complexity relative to SD-SDS by at least 75%, at low SNR, for 16-QAM. For 64-QAM, the detection complexity of the SD-SDS-DNN algorithm is at least 99% lower than that of SD-SDS at low SNR. This is all achieved whilst maintaining the BER performance close to that of SD-SDS and SD from literature. The SD-SDS-DNN algorithm lowers the detection complexity of Golden code to the point that it approaches the detection complexity of the Alamouti STBC linear ML detector for a spectral efficiency of 8 bits/s/Hz. For a spectral efficiency of 12 bits/s/Hz, the SD-SDS-DNN detection complexity is 90% lower than the Alamouti linear ML detector detection complexity.

In future research, the SD-SDS-DNN algorithm DNN channel condition predictor may be replaced with a low computational complexity random forest ensemble decision tree which may further reduce detection computational complexity in terms of number of FLOPS. Determination of the order of execution of the DNN detection algorithms in terms of inference time will be of value from a detection latency point of view.

## References

- [1]. S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journ. Sel. Areas Commun.*, vol. 16, no. 8, pp.1451-1458, 1998.
- [2]. H. Xu, K. Govindasamy, and N. Pillay, "Uncoded space-time labelling diversity," *IEEE Commun. Lett.*, vol. 20, no. 8, pp. 1511-1514, 2016.
- [3]. J. Belfiore, G. Rekaya and E. Viterbo, "The golden code: a 2 x 2 full-rate space-time block code with non-vanishing determinants," *International Symposium on Information Theory, ISIT 2004. Proceedings.*, Chicago, IL, 2004, pp. 310-310, 2004.
- [4]. A. Vielmon, Ye Li and J. R. Barry, "Performance of Alamouti transmit diversity over time-varying Rayleigh-fading channels," *IEEE Trans. on Wireless Commun.*, vol. 3, no. 5, pp. 1369-1373, 2004.
- [5]. M. O. Sinnokrot and J. R. Barry, "Fast maximum-likelihood decoding of the Golden code," *IEEE Trans. Wireless Commun.*, vol. 9, no. 1, pp. 26-31, 2010.
- [6]. N. Sharma, "Space Time Block Code for Next Generation Multi-user MIMO System," *9th International Conference on Future Networks and Communications, Procedia Computer Science*, pp. 172-179, 2014.
- [7] F. Riera-Palou and G. Femenias, "Improving STBC Performance in IEEE 802.11n Using Group-Orthogonal Frequency Diversity," *IEEE Wireless Communications and Networking Conference, Las Vegas, NV*, pp. 193-198, 2008.
- [8]. "IEEE Standard for Information technology--Telecommunications and information exchange between systems - Local and metropolitan area networks--Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Sub 1 GHz License Exempt Operation," in *IEEE Std 802.11ah-2016 (Amendment to IEEE Std 802.11-2016, as amended by IEEE Std 802.11ai-2016)*, pp.1-594, 2017.
- [9]. M. O. Sinnokrot and J. R. Barry, "Fast maximum-likelihood decoding of the Golden code", *IEEE Trans. Wireless Commun.*, vol. 9, no. 1, pp. 26-31, 2010.

- [10]. S. Kahraman and M. E. Çelebi, "Dimensionality reduced decoding for the golden code with the worst-case complexity of  $O(m^{1.5})$  for low range of SNR," IEEE Wireless Communications and Networking Conference (WCNC), 2012, pp. 246-250.
- [11]. S. Sirinaunpiboon, A. R. Calderbank, and S. D. Howard, "Fast essentially maximum likelihood decoding of the Golden code," IEEE Trans. Inf. Theory, vol. 57, no. 6, pp. 3537-3541, 2011.
- [12]. L. Zhang, B. Li, T. Yuan, X. Zhang, and D. Yang, "Golden code with low complexity sphere decoder," in Proc. 18th Int. Symp. Pers. Indoor Mobile Radio Commun., pp. 1-5, 2007.
- [13]. J. Jaldén and B. Ottersten, "On the complexity of sphere decoding in digital communications", IEEE Trans. Signal Process., vol. 53, no. 4, pp. 1474-1484, 2005.
- [14]. H. Xu and N. Pillay, "Reduced complexity detection schemes for Golden code systems," IEEE Access, vol. 7, pp. 139140-139149, 2019.
- [15]. H. Xu and N. Pillay, "Multiple complex symbol Golden Code," IEEE Access, vol. 8, pp. 103576-103584, 2020.
- [16]. M. Mohammadkarimi, M. Mehrabi, M. Ardakani and Y. Jing, "Deep Learning-Based Sphere Decoding," IEEE Trans. on Wireless Commun., vol. 18, no. 9, pp. 4368-4378, 2019.
- [17]. A. Askri and G. R. Othman, "DNN assisted Sphere Decoder," 2019 IEEE Int. Symposium on Inf. Theory (ISIT), Paris, France, pp. 1172-1176, 2019.
- [18]. N. T. Nguyen, K. Lee and H. Dai, "Application of Deep Learning to Sphere Decoding for Large MIMO Systems," in IEEE Transactions on Wireless Communications, DOI: 10.1109/TWC.2021.3076527.
- [19]. D. Weon and K. Lee, "Learning-aided deep path prediction for sphere decoding in large MIMO systems," IEEE Access, vol. 8, pp. 70870-70877, 2020
- [20]. B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," IEEE Trans. on Commun., vol. 51, no. 3, pp. 389-399, 2003.
- [21]. J. Li and Z. Wang, "An improved initial radius selection scheme for sphere decoding in MIMO", 3rd Int. Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB17).
- [22]. B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. Expected complexity", IEEE Trans. on Signal Processing, vol. 53, no. 8, pp. 2806-2818, 2005.

- [23]. J. Jalden and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Trans. on Signal Processing*, vol. 53, no. 4, pp. 1474-1484, 2005.
- [24]. F. Oggier, "On the optimality of the Golden code", in *Proc. IEEE Inf. Theory Workshop*, Punte del Este, Uruguay, pp. 468-472, 2006.
- [25]. O. A. Ivanova, *Unitary Matrix*, Encyclopedia of Mathematics, EMS Press, 1994.
- [26]. N. Samuel, T. Diskin, A. Wiesel, "Learning to detect" *IEEE Trans. on Signal Processing*, vol. 67, no. 10, 2019.
- [27]. D. B. J. Kingma, "Adam, a method for stochastic optimization", 2014.
- [28]. C. Candan, "Notes on Linear Minimum Mean Square Error Estimators", 2011.
- [29] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique", *Journal of Artificial Intelligence Research*, vol. 16, 2002.
- [30]. R. Y. Mesleh, H. Haas, S. Sinanovic, C. W. Ahn, and S. Yun, "Spatial modulation", *IEEE Trans. Veh. Technol.*, vol. 57, no. 4, pp. 2228-2241, 2008.

**Part III**  
**Concluding Remarks**

## Conclusion

Channel estimation is an important component of a wireless MIMO receiver. Our study concentrated on the training-based channel estimation methods which unfortunately come with a training sequence overhead that uses up expensive channel bandwidth. Because of this drawback of the training-based channel estimation, our research proposed a transmit power-sharing equivalent noise power optimization technique, for the USTLD-MIMO system, that reduced the number of required pilot training sequences, for channel estimation, without compromising the error rate performance. Our study also proposed the use of deep learning, for channel estimation, to eliminate the need to have prior knowledge of the training sequence matrix, average noise power and the wireless channel covariance matrix as needed by some of the traditional training-based channel estimators. The study concluded that a saving of 80% in the number of pilot training sequences, required to perform channel estimation without loss in error rate performance, was achieved for the case of 16-QAM and 16-PSK modulation in USTLD-MIMO. This saving was achieved relative to the traditional LS and MMSE channel estimators.

Golden code is a full-rate full-diversity two transmit antenna STBC system that can deliver both spatial multiplexing gain and higher link reliability in IoT based low power wireless MIMO links. Golden code can also help deliver high data throughputs and high link reliability that is a QoS requirement for the next generation wireless networks. However, the Golden space-time block code has the drawback of exponential detection complexity which may violate the ultra-low latency QoS requirement, especially for high-density M-QAM modulation, for next generation wireless networks. The exponential detection complexity also implies a very high number of FLOPS in decoding transmitted Golden code encoded M-QAM symbols. The high number of FLOPS required during decoding has a negative implication when it comes to receiver node power consumption, and this is not suitable for battery powered IoT devices. To meet the next generation wireless networks ultra-low latency QoS requirement and the low power wireless communication needs of IoT devices, our study proposed low complexity Golden code analytical and deep learning-based sphere-decoding algorithms. The proposed decoding algorithms achieved as much as 70% reduction in decoding latency, for the high-density M-QAM scenario, relative to the SD-SDS algorithm without loss in error rate

performance. For the reduction in number of detection FLOPS, and hence receiver power consumption, the proposed Golden code deep learning-based sphere-decoding algorithm achieved at most 99% reduction in detection complexity, at low SNR, relative to the SD-SDS algorithm whilst maintaining the error rate performance. The SD-SDS algorithm has been shown in literature to deliver lower Golden code detection complexity relative to the traditional SE-SD algorithm.

For future work, it will be interesting if the following scenarios could be investigated:

- The USTLD-MIMO bandwidth-efficient channel estimation, in this thesis, is optimized for a quasi-static Rayleigh fading wireless channel. It will be interesting to perform the optimization over a mobile wireless environment where the user is in motion relative to the RAN base station. The wireless fading channel becomes time-selective under the mobile conditions due to Doppler shift. It will be interesting to investigate if the number of pilot training sequences can be optimized for a time-selective wireless fading channel without loss in error rate performance.
- The USTLD-MIMO deep learning-based blind channel estimator, in this thesis, is trained using a wireless channel matrix that is assumed to be known without estimation errors. In this thesis, the MMSE wireless channel second order statistics are also assumed to be known without estimation errors. These assumptions are not valid for practical scenarios as the wireless channel and its covariance matrix will need to be estimated which inherently introduces errors. An investigation into the performance of a blind USTLD-MIMO deep learning-based channel estimator, trained using an estimated wireless channel, will need to be carried out. The deep learning-based wireless channel estimator performance needs to be compared to an MMSE channel estimator with an estimated wireless channel covariance matrix.
- The Golden code low complexity deep learning-based decoding algorithm developed to lower the number of detection FLOPS, uses a DNN architecture based on artificial neural networks (ANN) which rely on FLOPS arithmetic to perform predictions. It will be interesting to see if an ensemble random forest decision tree can be used for predictions instead of the FLOPS intensive ANN based architecture. An investigation could be carried out to find out if the ensemble random forest decision tree can further lower the

number of FLOPS required to perform the Golden code detection without loss in error rate performance.