12-8-2023

Neural Networks for improved signal source enumeration and localization with unsteered antenna arrays

John T. Rogers II
Mississippi State University, jtr2.primary@outlook.com

Neural Networks for improved signal source enumeration

and localization with unsteered antenna arrays

By

John T. Rogers II

Approved by:

John E. Ball (Major Professor)
Ali Gurbuz
Mehmet Kurum
Chaomin Luo
Qian (Jenny) Du (Graduate Coordinator)
Jason M. Keith (Dean, Bagley College of Engineering)

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Electrical and Computer Engineering
in the Department of Electrical and Computer Engineering

Mississippi State, Mississippi

December 2023

Name: John T. Rogers II

Date of Degree: December 8, 2023

Institution: Mississippi State University

Major Field: Electrical and Computer Engineering

Major Professor: John E. Ball

Title of Study: Neural Networks for improved signal source enumeration and localization with unsteered antenna arrays

Pages of Study: 79

Candidate for Degree of Doctor of Philosophy

Direction of Arrival estimation using unsteered antenna arrays, unlike mechanically scanned or phased arrays, requires complex algorithms which perform poorly with small aperture arrays or without a large number of observations, or snapshots. In general, these algorithms compute a sample covriance matrix to obtain the direction of arrival and some require a prior estimate of the number of signal sources. Herein, artificial neural network architectures are proposed which demonstrate improved estimation of the number of signal sources, the true signal covariance matrix, and the direction of arrival. The proposed number of source estimation network demonstrates robust performance in the case of coherent signals where conventional methods fail. For covariance matrix estimation, four different network architectures are assessed and the best performing architecture achieves a 20 times improvement in performance over the sample covariance matrix. Additionally, this network can achieve comparable performance to the sample covariance matrix with 1/8-th the amount of snapshots. For direction of arrival estimation, preliminary results are provided comparing six architectures which all demonstrate high levels of accuracy and demonstrate the benefits of progressively training artificial neural networks by training on a sequence of sub-problems and extending to the network to encapsulate the entire process.

Key words: Machine Learning, Neural Networks, Convolutional Neural Networks, Deep Learning, Number of Sources Estimation, Direction of Arrival Estimation, Covariance Matrix Estimation, Unsteered Antenna Arrays, RADAR

TABLE OF CONTENTS

LIST OF TABLES . . . v

LIST OF FIGURES . . . vii

LIST OF SYMBOLS, ABBREVIATIONS, AND NOMENCLATURE . . . ix

CHAPTER

I. INTRODUCTION . . . 1

1.1 Motivation . . . 1
1.1.1 Number of Sources Estimation . . . 2
1.1.2 Covariance Matrix Estimation . . . 3
1.2 Direction of Arrival Estimation . . . 4
1.3 Overview . . . 4

II. LITERATURE REVIEW . . . 5

2.1 Angular Localization . . . 5
2.2 Number of Sources Estimation . . . 6
2.3 Covariance Matrix Estimation . . . 7
2.3.1 Properties of the Eigen-Decomposition of the Auto-Covariance Matrix 9
2.4 Direction of Arrival Estimation . . . 10
2.4.1 Digital Beamforming . . . 11
2.4.2 MUSIC . . . 12
2.5 Neural Networks . . . 13
2.5.1 Covariance Matrix Estimation with Neural Networks . . . 13
2.5.2 Direction of Arrival Estimation with Neural Networks . . . 14

III. NUMBER OF SOURCES ESTIMATION NETWORK . . . 16

3.1 Introduction . . . 16
3.2 Background . . . 16
3.2.1 Conventional source estimation methods . . . 17
3.2.2 Coherent source estimation methods . . . 17

3.3 Methodology 17
3.3.1 Signal Model 18
3.3.2 Spatial Smoothing 20
3.3.3 Network Input Formatting 21
3.3.4 Radar parameters 23
3.3.5 Proposed Network 24
3.4 Data sets 25
3.5 Results and Discussion 26
3.5.1 Training Parameters 27
3.5.2 Test Case Analysis 27
3.6 Conclusions 34

IV. COVARIANCE MATRIX ESTIMATION NETWORK 36

4.1 Introduction 36
4.2 Background 36
4.2.1 Conventional Covariance Matrix Estimation 36
4.2.2 Array Imperfections 37
4.3 Methodology 37
4.3.1 Simulated Array Imperfections 38
4.3.2 Simulation Data 39
4.3.3 Utilized Neural Network Structures 41
4.3.4 Error Metrics 42
4.4 Results and Discussion 42
4.4.1 Evaluated Architectures 43
4.4.2 Final Network Results 44
4.4.3 Results with Simulated Array Imperfections 47
4.5 Conclusions 51

V. DIRECTION OF ARRIVAL ESTIMATION 53

5.1 Introduction 53
5.2 Background 53
5.3 Methodology 54
5.3.1 Network Architecture 55
5.3.2 Simulated Data 58
5.3.3 Error Metrics 58
5.4 Results and Discussion 59
5.4.1 Multiple Signal Sources 61
5.5 Comparison with similar Networks 65
5.6 Conclusions 65

VI. CONCLUSIONS AND FUTURE WORK 70

6.1 Discussion 70
6.2 Future Work 71

REFERENCES 73

LIST OF TABLES

3.1 Number of Sources Estimation Network Experimental training and testing scenarios 26

3.2 Number of Sources Estimation Network Training parameters 27

3.3 Proposed deep Number of Sources Estimation network architecture
PB = Processing Block: Fully Connected, Parametric ReLU, Batch Normalization,
DR = Dropout, SM = Softmax, CL = Classifier 28

3.4 Proposed Method Case A test confusion matrix 29

3.5 Proposed Method Case A overall results 29

3.6 Proposed Method Case B test confusion matrix 29

3.7 Proposed Method Case B overall results 29

3.8 Proposed Method Case C test confusion matrix 30

3.9 Proposed Method Case C overall results 30

3.10 Overall Number of Sources Estimation Network test results shown as accuracy in percent.
AIC, MDL, and EEF utilize spatial smoothing (The best results are in **bold**) 30

3.11 Comparison of Number of Sources Estimation Network test results for case B with different input combinations
C,SC = cov., smoothed cov. E,SE = eigenvalues cov. and smoothed cov.
C&E,SC&SE = cov. and eigs., smoothed cov. and eigs. OA = Overall Accuracy,
UE = Underestimated, OE = Overestimated (Best results in **bold**) 31
4.1 Covariance Matrix Estimation Network Architectures 44
4.2 Balanced Covariance Matrix Estimation Network Architectures 44
4.3 Covariance Matrix Estimation Network Architectures Results (Best results in **bold**) 45
4.4 Balanced Covariance Matrix Estimation Network Architectures Results (Best results in **bold**) 45
4.5 Finalized Covariance Matrix Estimation Network Architecture 46
4.6 Finalized Covariance Matrix Estimation Network Results 46
4.7 Results with Varied Number of Snapshots using Proposed Network Estimates and Sample Covariance Matrices 52
5.1 DOA Architectures' Results with 1 signal source (Best results in **bold**) 59
5.2 DOA Architectures' Results with $\pm 1^{\circ}$ Tolerance (Best results in **bold**) 60
5.3 DOA Architectures' Results with 1 to 4 signal sources (Best results in **bold**) 63
5.4 DOA Architectures' Results with 1 to 4 signal sources and a 0.25 threshold (Best results in **bold**) 64
5.5 Number of false alarms in DOA scatterplots 65

LIST OF FIGURES

3.1 Block diagram of Number of Sources Estimation Network data pre-processing stages 23

3.2 Accuracy of the proposed Number of Sources Estimation Network as a function of SNR 32

3.3 Accuracy of the proposed Number of Sources Estimation Network as a function of Number of Pulses 33

4.1 Proposed Covariance Matrix Estimation Architecture 38

4.2 Normalization Flowchart 41

4.3 Performance vs SNR for Network Results and Sample Covariance Matrix 46

4.4 Performance vs Number of Snapshots for Network Results and Sample Covariance Matrix 47

4.5 Normalized MUSIC Spectrum using the Network Results and Sample Covariance Matrix 48

4.6 Performance vs Phase Distortions 48

4.7 Performance vs Frequency Distortions 49

4.8 Performance vs Amplitude Distortions 49

4.9 Performance vs DC Offset Distortions 50

4.10 Performance with Differing IQ Distortions 51

5.1 DOA Network Component Architecture 56

5.2 Evaluated DOA Network Architectures 57

5.3 DOA Scatterplot with architecture 2. Standalone 61

5.4 DOA Scatterplot with architecture 5. CombStand 62

5.5 DOA Scatterplot with architecture 6, CombSeq, and 1 to 4 signal sources 62

5.6 DOA Scatterplot with architecture 1, Transfer Learning, with a 0.25 threshold and 1 to 4 signal sources 66

5.7 DOA Scatterplot with architecture 2, Standalone, with a 0.25 threshold and 1 to 4 signal sources 67

5.8 DOA Scatterplot with architecture 3, Sequence, with a 0.25 threshold and 1 to 4 signal sources 67

5.9 DOA Scatterplot with architecture 4, Combined, with a 0.25 threshold and 1 to 4 signal sources 68

5.10 DOA Scatterplot with architecture 5, CombStand, with a 0.25 threshold and 1 to 4 signal sources 68

5.11 DOA Scatterplot with architecture 6, CombSeq, with a 0.25 threshold and 1 to 4 signal sources 69

LIST OF SYMBOLS, ABBREVIATIONS, AND NOMENCLATURE

Acronyms

DOA Direction of Arrival Estimation

AOA Angle of Arrival Estimation

RADAR Radio Detection and Ranging

MSR Mechanically Scanned Radar

ESR Electronically Scanned Radar

FOV Field of View

MUSIC MUltiple SIgnal Classification

ESPRIT Estimation of Signal Parameters via Rotation Invariance Techniques

NN Artificial Neural Network

MLP Multi-layer Perceptron

CNN Convolutional Neural Network

MVDR Minimum Variance Distortionless Response

MLE Maximum Liklihood Estimator

OMP Orthogonal Matching Pursuit

RBFNN Radial Basis Function NN

AIC Akike Information Criteria

MDL Minimum Description Length

EEF Exponential Embedded Families

MIMO Multiple-Input Multiple-Output

ULA Uniform Linear Array

SNR Signal-to-Noise Ratio

SVD Singular Value Decomposition

FC Fully Connected (Perceptron) NN layer

ReLU Rectified Linear Unit

PReLU Parametric ReLU

SELU Scaled Exponential Linear Unit

BN Batch Normalization Layer

Conv Convolutional Layer

ConvT Convolutional Transpose Layer

MSE Mean Square Error

NMSE Normalized Mean Square Error

i.i.d. Independently and Identically Distributed

Symbols and Variables

$E[\bullet]$ Expected Value

$\bullet^H$ Hermitian Transpose

$\boldsymbol{X}$ Receiver input arranged by antenna element and pulse

x Single Element of $\boldsymbol{X}$

μ_x Mean of $\boldsymbol{X}$

$\boldsymbol{s}$ Noiseless Simulated RADAR Measurement

$\boldsymbol{K}$ Covariance Matrix

$\hat{\boldsymbol{K}}$ Estimated Covariance Matrix

$\boldsymbol{K}_s$ Sample Covariance Matrix

$\boldsymbol{K}_s^S$ Spatially Smoothed Sample Covariance Matrix

$\boldsymbol{R}$ Correlation Matrix

$\boldsymbol{\lambda}$ True Covariance Matrix Sorted Eigenvalue Vector

$\hat{\boldsymbol{\lambda}}$ Estimated Covariance Matrix Sorted Eigenvalue Vector

S Number of Receiver Element Subarrays used for Spatial Smoothing

$\boldsymbol{U}$ Eigenvector Matrix

$\boldsymbol{\Lambda}$ Diagonal Eigenvalue Matrix

M, m Number of Receiver Array Elements and Indexing Variable

P, p Number of Snapshots and Indexing Variable

T, k Number of Signal Sources and Indexing Variable

a Amplitude of Signal Source

f RADAR Operating Frequency

λ Wavelength corresponding to f

c Speed of Light in a Vacuum, $3 \cdot 10^8$

Δx Receiver Array Element Spacing

ϕ Angle of Signal Source

n Complex i.i.d. White Gaussian Noise Sample

λ, α SELU Constants

P_d Probability of Detection

P_m Probability of a Miss

P_{fa} Probability of a False Alarm

$P_d^{\pm 1^\circ}$ Probability of Detection with 1° tolerance

θ Direction of Steering Vector

$v(\theta)[i]$ Steering Vector Element for Direction θ and Array Element i

$\boldsymbol{v}(\theta)$ Steering Vector in Direction θ

CHAPTER I

INTRODUCTION

1.1 Motivation

In signal processing the general category of problems related to estimating the angles of targets or signal sources with respect to an array of sensors is named Direction of Arrival (DOA) estimation, or Angle of Arrival (AOA) estimation. In RADAR systems, DOA is achieved with one of three possible configurations. Mechanically Scanned RADARs (MSR) are physically rotating RADARs where arranging pulses according to the RADAR's direction can produce a power spectrum as a function of scanning angle. MSRs can produce 360° fine-angluar-resolution data that is simple to create, but the rotation introduces additional mechanical points of failure and imposes a relatively coarse temporal-angluar-resolution. Conversely, Phased Arrays are a form of Electronically Scanned RADARs (ESR) which uses phase shifts on the transmitted signal to perform beam steering. The traditional conformal array structures include planar, linear and circular arrangements of antenna elements. Like MSRs, Phased Arrays can directly compile the received signals into an angular product. As ESRs, Phased Arrays can have much finer temporal-resolution than MSRs, but the stationary nature of these RADARs can result a Field of View (FOV) less than 360° with many common configurations possessing FOVs less than 180°. Finally, RADARs can be utilize unsteered arrays which are the focus of this dissertation. Unsteered RADARs simplify antenna construction by removing the need for phase shifters or mechanical rotation, and the temporal-resolution is not dependant on scanning. However, these systems require more complex algorithms to extract angular information. For example, Digital Beamforming can be used to simulate the effects of beamsteering on the received data. While MSRs or Phased Arrays are generally preferable, unsteered RADAR processing is important for many array processing applications such as for low-power or repurposed RADAR systems as well as RADARs operating in either Passive or Signals of Opportunity modalities where manipulation of a transmitting antenna is not possible. Additionally, the underlying principles can be applied to other types of sensor arrays or extended to the larger problem of Blind Source Separation.

Associated with DOA estimation are the problems of Number of Sources estimation and Covariance Matrix Estimation. The use of Sample Auto-Covariance Matrix as an input for DOA algorithms is ubiquitous, thus improved estimation of the auto-covariance matrix can directly benefit these algorithms. Regarding Number of Sources estimation, some DOA algorithms depend on a prior estimate of the number of signal sources.

1.1.1 Number of Sources Estimation

The Number of Signal Sources, commonly referred to as targets, is an important metric for array processing. The number of sources is itself usable information by a radar operator in many situations such as indicating the amount of nearby objects of interest such as aircraft, vehicles, jammers, etc. Additionally, some DOA estimation techniques such as MUltiple SIgnal Classification (MUSIC) [61] and Estimation of Signal Parameters via Rotation Invariance Techniques (ESPRIT) [58] require an accurate estimate of the number of signal sources as part of the algorithms' computation.

The most common technique for Number of Sources estimation is the Akaike Information Criterion [2] which is applied to the eigen-values of the covariance matrix. In the area Number of Sources Estimation, an Artificial Neural Network (NN) is proposed herein which contributes:

- A robust NN system that achieves state-of-the-art results and surpasses traditional eigenvalue-based methods, which fail when the received data is coherent.
- Fusion of the spatially smoothed covariance matrix and the eigenvalues for joint analysis in a unified NN structure.
- Enhanced results even under small number of receivers or pulses in a coherent processing interval.

1.1.2 Covariance Matrix Estimation

Many DOA algorithms utilize the Sample Auto-Covariance Matrix of the received signal as an input. The eigen-decomposition of the auto-covariance matrix separates the information related to each signal source which is further detailed in section 2.3.1. Both DOA and Number of Sources estimation algorithms exploit these properties, thus these algorithms benefit from more accurate estimates of the covariance matrix.

The sample covariance matrix is the most common covariance matrix approximation. For improved estimation, there exists family of algorithms [35, 36, 4, 80, 40, 29] which modify the eigen-values of the sample covariance matrix to conform to the expected structure detailed in section 2.3.1. Herein, a NN for auto-covariance matrix estimation is proposed. The contributions to the area of Auto-Covariance Matrix Estimation presented herein include:

- A comparison of four covariance matrix estimation network architectures.
- Significantly improved estimation performance compared to the sample covariance matrix.
- A constrained network size allowing for efficient computation.

1.2 Direction of Arrival Estimation

Conventional techniques for DOA Estimation includes digital beamforming techniques and sub-space techniques. Digital Beamforming algorithms, such as the Minimum Variance Distortionless Response (MVDR) [9], performs an optimized projection of the auto-covariance matrix across an angular spectrum. Sub-space algorithms, such as MUSIC [61], extends digital beamforming by using eigen-decomposition to separate the noise and signal sub-spaces before projecting accross an angular spectrum. Herein, multiple NN architectures are proposed for DOA Estimation which contribute:

- An extension of the Covariance Matrix Estimation Network applied to DOA Estimation using a classification network formulation.
- A comparison of six proposed network architectures for DOA Estimation.
- Preliminary results demonstrating accurate DOA estimates without the need for computationally inefficient matrix operations.

1.3 Overview

This dissertation is organized in six chapters. Following this chapter is a review of the literature for Number of Sources Estimation, Covariance Matrix Estimation, and Direction of Arrival Estimation. The chapters 3-5 detail the contributions to Number of Sources Estimation, Covariance Matrix Estimation, and Direction of Arrival Estimation respectively. The final chapter includes overall conclusions and suggestions for future work.

CHAPTER II

LITERATURE REVIEW

2.1 Angular Localization

Classical RADAR processing focuses on detection and localization relative to radial distance, however, angular detection and localization is needed to find targets or other signal sources in 3D space. MSRs obtain this information by physically rotating the sensor while ESRs use phase shifts to move the beam with a stationary sensor array. Both of these techniques depend on steering of the transmitted beam, but angular localization is still possible with a stationary beam if there are multiple receiver elements. As receiver elements in an array are not perfectly co-located, the radial distance from a signal source to each element will slightly differ resulting in a phase shift in the reflected signal. This results in a sinusoid projected across the array with the arrangement of array elements corresponding to the sampling. In the case of a Uniform Linear Array (ULA) the ideal, projected sinusoid is given by equation 2.1:

$$x_m = \sum_{k=1}^{T} a_k \exp\left(-j2\pi \frac{f}{c} \Delta x\, m \sin\left(\phi_k\right)\right) \tag{2.1}$$

where x_m is the power measured by the m-th receive array element, a_k is the amplitude of the k-th signal source, T is the true number of sources, $j = \sqrt{-1}$, f is the operating frequency in hertz, c is the speed of the light in a vacuum in meters per second, Δx is the spacing between array elements in meters, m is the array element index, and ϕ_k is the angle of the $k-th$ signal source in radians.

For unsteered arrays there are two primary estimation problems for angular localization. First is angular detection which corresponds to the estimation of the Number of Signal Sources. Second is the estimation of the Direction of Arrival (DOA) for each signal source. Related to these problems is the estimation of the Auto-Covariance Matrix for this signal as the use of the covariance matrix as an input for Number of Sources and DOA Estimation algorithms is ubiquitous.

2.2 Number of Sources Estimation

Estimating the number of plane wave sources is an important problem in fields such as radar, sonar, and communication systems. In radar signal processing, estimating the number of signals present in noisy data is a difficult problem which has been explored extensively. However, robust solutions in this area are still required. It is advantageous for the radar to know how many sources are present in a signal to facilitate improved target detection and tracking. Traditional approaches include the Akike Information Criteria (AIC) estimator [2] and Minimum Description Length (MDL) estimator [54] which exploit the structure of the eigenvalues of the sample covariance matrix. This dependence exclusively on the eigenvalues however causes these methods to fail when the received signals are coherent. Additionally, the MVDR can be utilized to both estimate DOA and to estimate the number of sources, as long as the sources are separated adequately.

The AIC [2] is arguably the most widely used method for number of sources estimation in the case of white Gaussian noise. It determines the number of sources by minimizing a criterion over a range of detectable number of sources. However, the AIC is observed to provide inconsistent estimates and often overestimates the number of signals in radar applications [54]. As an alternative, the MDL was proposed in [54], but the MDL can lead to an underestimate of the signal subspace dimension, most commonly when the number of samples are comparably small [53]. Another model order criterion applied to source estimation is the Exponential Embedded Families (EEF) criterion [78]. This method has been demonstrated to outperform the MDL in difficult scenarios such as those with low SNR, closely spaced targets, and a limited number of signals.These methods operate on the eigenvalues of the signal covariance matrix to estimate the dimensionality of the data. More accurate methods have been proposed such as in [53] which uses a discriminate function on the covariance eigenvalues to estimate the dimensionality of both the signal and noise subspaces. These estimates are combined in another discriminate function to estimate the number of sources.

The techniques mentioned in here are designed for and work well when data is incoherent. It is observed that the performance of both number of source estimation and DOA estimation degrades when presented with coherent data [39, 49]. Additional pre-processing, such as spatial smoothing, is required for coherent signals [62]. The processing of coherent signals has been explored in [8, 41, 15, 24, 31, 30, 39, 51, 52, 63, 62, 71, 75, 77, 81].

Additionally, methods making the assumption that the number of sources is known *a priori* may give misleading results if the assumed number of sources is wrong [37]. In fact, the capability of resolving two closely spaced sources by an array of sensors is limited by its ability to estimate the number of sources correctly. If the number of sources can be determined more reliably, then fine-resolution sub-space DOA estimation methods can be applied more effectively.

2.3 Covariance Matrix Estimation

Covariance matrices have many desirable properties which has led to their use as inputs for many signal processing algorithms. A covariance matrix consists of the pairwise covariances between two subsets of random variables. An auto-covariance matrix is the covariance matrix of a set of random variables with itself and it is widely used in many communications, radar, and array signal processing applications. The auto-covariance matrix is defined in equation (2.2), where $\boldsymbol{\mu}_x$ is a vector of the means for each random variable in the set of random variables $\mathbf{x}$, and $\bullet^H$ is the Hermitian Transpose When the random variables are zero mean, this equation simplifies to equation for the auto-correlation matrix. Herein, all covariance matrices are zero-mean, auto-covariance matrices.

$$\mathbf{K} = E[\mathbf{x} - \boldsymbol{\mu}_x]E[\mathbf{x} - \boldsymbol{\mu}_x]^H = E[\mathbf{x}\mathbf{x}^H] - \boldsymbol{\mu}_x\boldsymbol{\mu}_x^H \tag{2.2}$$

As the true covariance matrix is often not known in practice, the sample covariance matrix is generally used as an approximation. The sample covariance matrix approximation for zero-mean data is presented in equation (2.3) where P is the number of snapshots.

$$\mathbf{K}_s = \frac{1}{P}\mathbf{X}\mathbf{X}^H \tag{2.3}$$

Improved estimation of the auto-covariance matrix, compared to the sample covariance, is an open area in signal processing with many recent developments. Knowledge of the application scenario, such as the array configuration, can be used to achieve improved covariance matrix estimation. In [80], interference steering vectors are used to reconstruct the interference plus noise covariance matrix. Upadhya and Vorobyov [72] presented an algorithm for covariance matrix estimation for Multiple-Input Multiple-Output (MIMO) systems. Deep neural networks are used in [45] for covariance matrix estimation in a computer vision application. In [4], covariance matrix estimation is modeled as an optimization problem using geometric considerations. In [35, 36], Kang et al. present a Rank-Constrained algorithm for Covariance Matrix Estimation.

The underlying principle of conventional Covariance Matrix Estimation algorithms such as [35, 36, 4, 80, 40, 29], is to improve the estimate of the eigenvalues by using a prior estimate of the number of sources to force the eigenvalues to conform to the ideal structure discussed in section 2.2. However, these techniques have no benefit to algorithms such as MUSIC which rely solely on the eigenvectors. Furthermore, the eigen-values are related signal power, which is explained in section 2.3.1, but the signal source's direction of arrival is related to the frequency information contained with the unmodified eigen-vectors.

2.3.1 Properties of the Eigen-Decomposition of the Auto-Covariance Matrix

Most of the aforementioned algorithms exploit the desirable properties of the Auto-Covariance Matrix. As the individual receiver array elements vary slightly in relative distance to a measured signal source, there are variations in the phase of the measured signal. This results in a series of sinusoids, one for each signal source, projected across the array with the array geometry determining the sampling. The simplest geometry, the Uniform Linear Array (ULA), equates to uniform sampling. The frequency of the projected sinusoids are related to relative angle between the signal source and the receiver. This is discussed further in section 3.3.1.

As sinusoids of different frequencies are orthogonal, the eigen-decomposition of the auto-covariance matrix separates this sinusoids except in the case of coherent signals. The resulting eigen-vectors correspond to these sinusoids with their corresponding eigen-values equaling the sum of the signal and noise power. Of course, the number of signal sources is not always equal to the number of receivers, thus the remaining eigen-vectors are related to the noise and their eigen-values are equal to the noise power.

This behavior is distorted when using an estimated covariance matrix such as the sample covariance matrix; thus, techniques like those presented in section 2.2 are required to estimate the Number of Signal Sources from estimated eigen-values by trying to find a similarly valued group among the smallest eigen-values. Similarly, the MUSIC algorithm exploits this structure by using an estimate of the Number of Signal Sources, to separate the signal and noise subspaces.

2.4 Direction of Arrival Estimation

Conventional DOA Estimation techniques include digital beamforming and many related techniques. Beamforming techniques such as the Minimum Variance Distortionless Response (MVDR) [9] apply steering vectors to the covariance matrix of a signal to project the signal power across an angular spectrum. Subspace algorithms exploit the desirable properties of the eigen-decomposition of the covariance matrix as the signals and noise are separated into orthogonal subspaces. The MUSIC algorithm [61] applies beamforming to eigen-vectors corresponding to the noise subspace. While subspace methods such as MUSIC or ESPRIT require an estimate of the number of signal sources, other techniques such as MVDR can produce an angular power spectrum with beamforming alone. These techniques specifically are called super resolution techniques as they can localize more accurately than the Rayleigh Resolution [73]. Moreover, these superresolution techniques, in general, require extensive computations and are generally too slow for real-time implementation [17]. The Maximum Likelihood Estimator (MLE) is an accurate DOA estimation method, but even its efficient implementation [82] can be too computationally burdensome. The

MLE requires a prior estimate of the number of sources and like MUSIC or MVDR it requires a parameter sweep. Its computational complexity grows exponentially with dimension.

More recently, approaches based on spatial sparsity of angle domain using compressive sensing [64, 23], smoothed-ℓ_0 norm [44], sub-Nyquist sampling [46] or quadrilinear decomposition [76] have been successfully applied to DOA estimation problems. Sparsity based DOA approaches solve constrained optimization problems or apply greedy approaches such as Orthogonal Matching Pursuit (OMP) [70]. Selections of constraints or stopping criteria for these techniques either require knowledge of the number of sources or good estimation of noise level in the measurements.

2.4.1 Digital Beamforming

Classical approaches to DOA Estimation use beamforming to project the signal power across an angular spectrum. These techniques rely on steering vectors constructed from the geometry of the array. For a 1-D array, the steering vector constructed with equation 2.4 where θ is the angle on which the covariance matrix is being projected, $-j$ is $\sqrt{-1}$, i is the receiver element index, d is the spacing between receiver elements in meters, and λ is the wavelength in meters corresponding to the operating frequency of the array.

$$v(\theta)[i] = \exp\left(-j2\pi i\frac{d}{\lambda}\sin\theta\right) \tag{2.4}$$

A large variety of beamforming techniques exist which apply the steering vector to project onto a given angle. An example is the MVDR [9] which minimises the noise variance in the projected direction. The MVDR beamforming weights are given by equation 2.5, from which the projected power measurement in direction θ can be obtained as $\boldsymbol{W}_{MVDR}^{H}(\theta)\boldsymbol{K}_s\boldsymbol{W}_{MVDR}(\theta)$, where

$\boldsymbol{K}_s$ is the Sample Covariance Matrix and $\boldsymbol{v}(\theta)$ is the steering vector in direction θ. An angular power spectrum can be produced by evaluting these equations for multiple values of θ with the location of signal sources appearing as peaks in the angle spectrum.

$$\boldsymbol{W}_{MVDR}(\theta) = \frac{\boldsymbol{K}_s^{-1}\boldsymbol{v}(\theta)}{\boldsymbol{v}(\theta)^H\boldsymbol{K}_s^{-1}\boldsymbol{v}(\theta)} \tag{2.5}$$

2.4.2 MUSIC

The MUSIC [61] algorithm is similar to beamforming techniques, but first separates the signal and noise subspaces. By using a prior estimate of the Number of Signal Sources, T, the signal subspace can be found using the aforementioned properties of the eigen-decoposition of the Auto-Covariance Matrix. Specifically, the T largest eigen-values and their corresponding eigen-vectors correspond to the signal subspace with the remaining set corresponding to the noise subspace. The MUSIC algorithm constructs an approximate covariance matrix of noise subspaces as product of the matrix constructed from the noise eigen-vectors and the Hermitian transpose of this matrix. By using the steering vector presented in equation 2.4, this noise matrix is projected across an angular spectrum, producing minima at the angles corresponding to the signal sources; however, the spectrum is typically inverted to produce maxima at the target angles like the spectrums from beamforming techniques. Thus, the equation for a sample of a MUSIC spectrum is usually given as shown in equation 2.6 where $\boldsymbol{U}_n$ is the matrix constructed from the noise eigen-vectors.

$$MUSIC(\theta) = \frac{1}{\boldsymbol{v}(\theta)^H\boldsymbol{U}_n\boldsymbol{U}_n^H\boldsymbol{v}(\theta)} \tag{2.6}$$

2.5 Neural Networks

Artificial Neural networks (NNs) are a family of machine learning function approximation techniques loosely meant to mimic neurons in the brain. Traditional NNs are known as Multi-layer Perceptrons (MLP) which arrange a few layers of perceptron functions and non-linear activation functions to allow for the approximation of non-linear functions. It has been proven that a two-layer MLP is sufficient to approximate any function [12], but number of perceptrons required is often too computationally burdensome. The introduction of batch normalization has allowed for stable networks with more layers. It has been observed that networks with more layers can often achieve superior results with a smaller overall structure than a two-layer MLPg. Additionally, Deep Learning (DL) has enabled practical use of convolutional layers leading to spatially-focused Convolutional Neural Networks (CNN) becoming one of the most common NN architectures.

2.5.1 Covariance Matrix Estimation with Neural Networks

The estimation of the Auto-Covariance Matrix with NNs for array signal processing has not been widely explored. Some work has been performed for estimating the channel covariance [42, 79] for communications applications. Hoffbeck, et al. [29] presented a non-NN, learning-based algorithm for covariance estimation. NNs have also been used to estimate covariance matrices in the area of finance [13, 43]. In [6], a neural network is used to estimate the covariance matrices of antenna array subsets on which MUSIC is applied to estimate the DOA.

Many proposed covariance estimation algorithms are only tested using hundreds of snapshots which corresponds to an unreasonable time delay for real-time operation.

2.5.2 Direction of Arrival Estimation with Neural Networks

Several works have developed NN approaches to DOA estimation in radar over the past 30 years [33, 48] using a wide variety of NN architectures. In [1], a classical 3-layer Multi-Layer Perception (MLP) NN is applied to 2D DOA estimation. El Zoohgby et al. [16, 17, 18] utilized the covariance matrix and a radial basis function (RBF) NN to estimate the DOA of multiple radar signals. RBFNNs were also used in [48, 69, 25, 34, 59, 60, 48]. [49] also uses a RBFNN and focuses on estimation of the DOAs for two signal sources whereas many DOA estimation algorithms focus on estimation for a single signal source. Kim, et al. [37] apply NN to estimate the DOA of human targets with a small antenna array unlike most algorithms which focus on large arrays. In [65], a Fuzzy NN was applied to DOA estimation using the phase difference as an input. Du et al. [14] examines several NN architectures for antenna array signal processing: multi-layer perceptrons, Hopfield networks, Radial-Basis Function NN (RBFNN), PCA-based NN, and Fuzzy NN. In [20], a NN-based DOA Estimation algorithm is embedded into a compact digital signal processing module demonstrating the integration of these algorithms in classical hardware, however this device only has an FOV of 45°. A DOA estimation NN is extended to angular tracking in [11]. In [68] a NN is proposed for adaptive beamforming which can adapt to failures and imperfections in a phase array.

NN have also been applied in the area of Blind Source Separation of which DOA estimation is a subproblem. Amari and Cichocki [3] examined adaptive blind signal processing using NN and provided a list of ten open questions in the field. Solazzi et al. [67] developed a spline NN to address blind source separation. These networks are not directly applicable for radar as the radar data is complex. Complex NNs have been studied for about 20 years now [27, 28], but there is little published on radar processing using a complex NNs.

Deep Learning (DL) has gained much attention in various research communities due to significant performance gains of many DL systems over more standard (hand–crafted) feature based learning. Deep networks can learn very complicated features and decision boundaries directly from the raw data. Grais et al. [22] utilized a deep (five layer) NN where the initial estimates were generated using non-negative matrix factorization. Their system identified the data source (source one or source two) in speech processing. Vesperini et al. [74] put forth a DL system that could handle both static and moving sound sources. In [47] a sequence of an auto-encoder and a set of classification networks is proposed for DOA Estimation which is robust to array imperfections.

Among the state-of-the-art techniques for DOA Estimation is the deep CNNs presented in [19, 50]. This CNN utilizes a Binary-Cross Entropy Loss function to train the network as a classification problem with the output being the angle spectrum. This network achieves accurate results but is only evaluated on data with thousands of snapshots which is not practical for most real-time applications.

Most of the proposed DOA algorithms exhibit one or more of the following issues when evaluating the algorithms performance: the algorithm only estimates single signal source's angle, the antenna array used is very large which is not true for all antennas, or hundreds snapshots are used to construct the sample covariance matrix which imposes a significant time delay in real-time operations.

CHAPTER III

NUMBER OF SOURCES ESTIMATION NETWORK

3.1 Introduction

This chapter introduces a NN architecture for estimating the Number of Signal Sources with an unsteered antenna array. This work was published in the following:

- J. Rogers, J. E. Ball, and A. C. Gurbuz, "Estimating the Number of Sources via Deep Learning," *2019 IEEE Radar Conference (RadarConf). IEEE,* 2019, pp. 1–5.
- J. Rogers, J. E. Ball, and A. C. Gurbuz, "Robust estimation of the number of coherent radar signal sources using deep learning," *IET Radar, Sonar & Navigation*, vol. 15, no. 5, 2021, pp. 431–440.

3.2 Background

Estimating the number of plane wave signal sources is an important problem in fields such as radar, sonar, and communication systems. An accurate estimation of the number of signal sources is required by many DOA algorithms. Conventional Number of Sources Estimation techniques exploit the properties of the eigen-decompostion described in section 2.3.1 to define the number of signal sources as the number of non-equal eigenvalues. However, the eigenvalues corresponding to the noise subspace are only equal in the ideal case, thus metrics such as the Akaike Information Criterion (AIC) [2] and Minimum Description Length (MDL) [54] are required. As a more accurate estimate of the covariance matrix will more closely conform to the desired eigenvalue structure, these techniques can directly benefit from improved covariance matrix estimation.

3.2.1 Conventional source estimation methods

The AIC[2] is minimized over a range of detectable number of sources to find the most likely estimate. However, the AIC is observed to provide inconsistent estimates and often overestimates the number of signals in radar applications [54]. An alternative, the MDL was proposed in [54], but this algorithm tends to underestimate of the signal subspace dimension, most commonly when the number of samples are comparably small [53]. An additional model order criterion, the Exponential Embedded Families (EEF) criterion[78] has been proposed for number of sources estimation. This method has been demonstrated to outperform the MDL in difficult scenarios such as those with low SNR, closely spaced targets, and a limited number of signals. All of these algorithms operate solely on the eigenvalues of the signal covariance matrix to estimate the dimensionality of the data.

3.2.2 Coherent source estimation methods

The aforementioned techniques are designed for and work well when data is incoherent. It has been observed that the performance of both number of source estimation and the DOA estimation degrades in the presence of coherent data. Additional pre-processing, such as spatial smoothing, is required for processing coherent signals [62]. The most common approach to separating coherent signal source is spacial smoothing which is described in section 3.3.2.

3.3 Methodology

Herein, a NN for number of source estimation is presented. This work focuses on the number of sources in coherent signals and utilizes spatial smoothing on the input sample covariance matrices. The proposed NN utilizes inputs constructed from the separated real and imaginary values of the covariance matrix and its eigenvalues. The proposed network attempts to fuse these inputs in an optimal manner to give a robust estimator of the number of sources in a signal. The network proposed herein is associated with the work presented in [55, 56].

3.3.1 Signal Model

All the methods proposed herein utilize synthetic data created by simulating a Uniform Linear Array (ULA) according to the following:

$$x_m = \left[\sum_{k=1}^{T} a_k \exp\left(-j2\pi \frac{f}{c} \Delta x \, m \sin\left(\phi_k\right) \right) \right] + n_m \tag{3.1}$$

where x_m is the power measured by the m-th receive array element, a_k is the amplitude of the k-th signal source, T is the true number of sources, $j = \sqrt{-1}$, f is the operating frequency in hertz, c is the speed of the light in a vacuum in meters per second, Δx is the spacing between array elements in meters, m is the array element index, ϕ_k is the angle of the $k - th$ signal source in radians, and n_m is independent and identically distributed (i.i.d) complex white Gaussian noise observed by the $m - th$ array element. A single array measurement is simulated by arranging x_m measurements for all M receive elements as a vector with randomly generated complex noise. The final signal is constructed by arranging P snapshots of the array measurement into a $[M \times P]$ matrix $\mathbf{X}$.

These simulated signal matrices are further processed to obtain the sample and true covariance matrices. As the signal is sinusoidal and the noise is white, the simulated signal matrices have a true mean of 0, thus the covariance matrix is equal to the correlation matrix. Therefore, the sample covariance matrix, $\mathbf{K}_s$ can be calculated as the approximated correlation matrix defined as follows:

$$\mathbf{K}_s = \frac{1}{P} \mathbf{X} \mathbf{X}^H \tag{3.2}$$

Likewise the true covariance matrix is equal to the true correlation matrix given by:

$$\mathbf{K} = E[\mathbf{X}\mathbf{X}^H] \tag{3.3}$$

As the noise is independent for each receiver array element, the true covariance between different elements is unaffected by noise. As the covariance matrix diagonal contains the variance for each element, these values are effected by noise . The diagonal terms are equal to the diagonal terms of the corresponding noiseless covariance matrix plus the noise variance. Therefore, the true covariance matrix can be found using (3.4) where **s** is a vector containing a noiseless measurement of the signal constructed using only the bracketed portion of (3.1), σ_n^2 is the noise power, and **I** is the identity matrix.

$$\mathbf{K}_s = \mathbf{s}\mathbf{s}^H + \sigma_n^2\mathbf{I} \tag{3.4}$$

To adjust the coherency between signal sources, the data is first made incoherent by simulating the response for each signal source separately and inducing random phase shifts. These vectors are then combined into a matrix. Then a coherency matrix is constructed such that the off-diagonal terms correspond to the level of coherency between each pair of targets with the main diagonal equaling one. This Hermitan matrix is separated and multiplied by the aforementioned matrix of signal source response vectors. The off-diagonal coherency terms range from 0 to 1 with all 0's resulting in an identity matrix preserving the incoherence introduced by the random phase shifts. Conversely, all 1's produces a matrix of ones which equalizes the phase shifts creating fully coherent signals.

3.3.2 Spatial Smoothing

Spatial smoothing [62] utilizes subarrays to separate the signal information. This method divides the receiver array into multiple overlapping subarrays of equal size. The signal covariance matrix is calculated for each subarray and averaged to produce a single covariance matrix for the signal. Variations on spatial smoothing include Du and Kirlin's method [15] which considers the cross–correlation of the subarray outputs and Hung and Kaveh's method [30] which uses so–called "focussing matrices". Another method employs an exchange matrix with correlated/coherent signals content and then uses a search over the parameter space and peak–finding [52].

Let a radar receiver have M channels, and define the $[M \times P]$ receiver data matrix as $\mathbf{X}$, where M is the number of reciver elements and P is the number of pulses. To estimate the non-spatially smoothed (e.g. standard) covariance matrix, one can use

$$\mathbf{K_s} = \frac{\mathbf{1}}{\mathbf{P}}\mathbf{X}\mathbf{X}^{\mathbf{H}} \tag{3.5}$$

where the Hermitian matrix transpose is denoted by $(.)^H$. The covariance matrix will be sized $[M \times M]$. To estimate the smoothed covariance matrix, the receiver is divided into S-element subarrays, where the m-th subarray corresponds to receiver elements $\{m, m+1, \cdots, m+S-1\}$, and let the m-th subarray data be $\mathbf{X}_m$. The spatially smoothed subarray covariance matrix will have size $[S \times S]$ and is calculated as

$$\mathbf{K}_s^S = \frac{1}{(M-S+1)\,P} \sum_{m=1}^{M-S+1} \mathbf{X}_m (\mathbf{X}_m)^H \tag{3.6}$$

where spatial smoothing is indicated by the superscript S.

3.3.3 Network Input Formatting

In order to estimate the number of sources, a combination of the estimates of the covariance matrix itself and the covariance matrix eigenvalues is used herein. For each case analyzed, the $[M \times M]$ covariance data matrix is estimated using the same spatial smoothing averaging methods used in [77] and [62].

Herein, the spatially–smoothed covariance matrix is unrolled to produce a $\left[2M^2 \times 1\right]$ feature vector as follows, where $M = S$:

$$\mathbf{f}_R = \begin{bmatrix} real\left\{\mathbf{K}_s^S\left(1,1\right)\right\} \\ real\left\{\mathbf{K}_s^S\left(2,1\right)\right\} \\ \vdots \\ real\left\{\mathbf{K}_s^S\left(M,1\right)\right\} \\ \vdots \\ real\left\{\mathbf{K}_s^S\left(M,M\right)\right\} \\ imag\left\{\mathbf{K}_s^S\left(1,1\right)\right\} \\ imag\left\{\mathbf{K}_s^S\left(2,1\right)\right\} \\ \vdots \\ imag\left\{\mathbf{K}_s^S\left(M,1\right)\right\} \\ \vdots \\ imag\left\{\mathbf{K}_s^S\left(M,M\right)\right\} \end{bmatrix}. \tag{3.7}$$

Although the diagonal terms of the covariance matrix should be real, for simplicity, all entries of the covariance matrix is unrolled for the feature vector. Thus, the omission of zero imaginary parts of diagonal elements is an opportunity for optimization not implemented here. The eigenvalues are computed using the Singular Value Decomposition (SVD) [5] as follows from the covariance matrix:

$$\mathbf{K}_s^S = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^H, \tag{3.8}$$

where $\mathbf{\Lambda}$ is the diagonal singular–value matrix whose diagonals are the covariance matrix eigenvalues: $\mathbf{\Lambda} = diag\left(\lambda_1^S, \lambda_2^S, \cdots, \lambda_M^S\right)$, where $\lambda_1^S \geq \lambda_2^S \geq \cdots \geq \lambda_M^S$. The eigenvalue features are placed in the $[M \times 1]$ feature vector as follows

$$\mathbf{f}_{\lambda^S} = \left[\lambda_1^S, \lambda_2^S, \cdots, \lambda_M^S\right]^T. \tag{3.9}$$

The final data feature is the $\left[\left(2M^2 + M\right) \times 1\right]$ vector given by

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_R \\ \\ \mathbf{f}_{\lambda^S} \end{bmatrix}. \tag{3.10}$$

Whereas, the smoothed covariance matrix and its eigenvalues were the primary inputs used in the experiments, the unsmoothed covariance matrix estimate $\mathbf{K}_s$ and its eigenvalues λ were also tested as network inputs. Figure 3.1 shows the data preprocessing stages to produce the four input data sources. In this figure, the ”Input Select and Formatting” block selecting combinations of the input data sources and converting these to a single vector as discussed previously in this section. Herein, six combinations were tested as network inputs. All four inputs were tested independently as well as each covariance matrix paired with its corresponding eigenvalues.

3.3.4 Radar parameters

The dataset utilized herein simulates a radar consisting of a notional uniform linear array operating at $f = 5.0$ GHz with element spacing $\Delta x = \lambda/2$. There are $M = 11$ receivers in the array, and this value was chosen to present a small array (in order to challenge the algorithms). The number of pulses per coherent processing interval is $P = 10$, which was also chosen to be a smaller number of pulses in order to evaluate the proposed method with a smaller number of pulses. The subarray size used in spatial smoothing is $S = 6$.

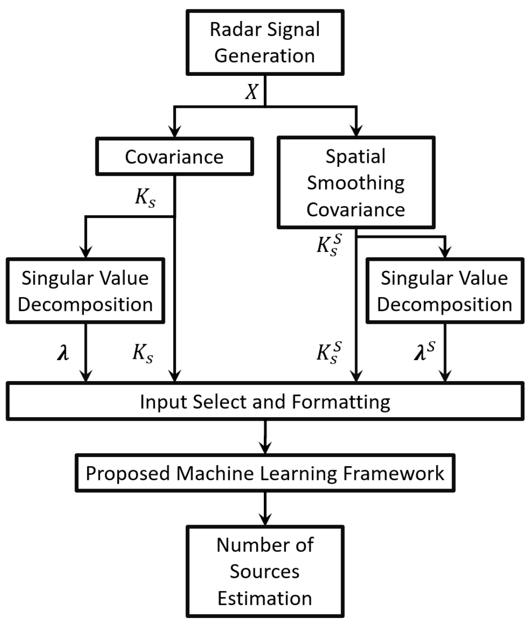

Figure 3.1

Block diagram of Number of Sources Estimation Network data pre-processing stages

3.3.5 Proposed Network

The network proposed herein produces robust results with a small amount of neurons. In the case of receiver with 11 channels utilizing subarrays of six elements each for spatial smoothing, the network will have 78 input features. The proposed network has nine fully–connected (FC) layers, a parametric rectified linear unit (PReLU) and a batch normalization layer.

The PReLU [26] is a non-linear operator that allows both positive and negative values to pass through the layer. Improved performance was observed compared to a similar network using rectified linear unit (ReLU) activations, which do not allow negative inputs to pass through. The PReLU is defined as

$$f(x) = \begin{cases} x & x \geq 0 \\ -\alpha x & x < 0 \end{cases}, \tag{3.11}$$

where α is a network-learned parameter.

To mitigate overfitting (that is, learning the training data at high accuracy but not being able to generalize well to novel data), a dropout layer is used. After the last FC layer, a softmax and classifier layer provide the final outputs. During training, the dropout layer randomly selects 50% of the outputs connections to be set to 0. During testing, all outputs are available. The FC layers have a connection to each output of the previous layer. They also have an activation function and compute their output as the dot product of the input vector with the internal weight vector plus the bias term. Then the linear or non-linear activation function is applied to provide the final FC output. The batch normalization layers learn input data distribution mean and variance and normalize the output by making it tend to zero mean and unity variance. Batch normalization layers are utilized to allow for a larger amount of network layers [32]. Finally, the softmax layer in conjunction with the classification layer learns a distribution to estimate the number of sources [7].

Herein, the network weights are initialized randomly from Gaussian distribution with zero mean and variance 0.01, and the biases are initialized as zero, and the network weights are updated using stochastic gradient descent with momentum [7]. The NN architecture is described in Table 3.3. The sizes of the FC layers are integer multiples of the input vector size.

3.4 Data sets

The network presented herein is tested on simulated data and compared to contemporary methods. Table 3.1 shows the various train and test cases utilized. In Table 3.1, the function $rand\,[A, B]$ represents the selection of a random value from a uniform distribution in the range $A \leq x \leq B$. The SNR values in Table 3.1, are independent for each signal.

Each signal features zero to three sources. If no source is simulated, the signal contains only noise. The angles of sources are selected randomly from a uniform distribution within a limited FOV of $-60°$ to $60°$. A minimum source spacing is enforced in the case of multiple sources as sufficiently close sources are impossible to distinguish. This minimal spacing is set to $\Delta\phi = 0.5°$ for all cases. With these parameters, coherent signals are simulated and added together with i.i.d complex WGN. To simulate coherent sources first independent signals are generated and Cholesky decomposition of the covariance matrix with desired coherency level σ is used to generate coherent signals as detailed in [21, 66].

Test case A has all targets at 10 dB SNR. This test case might be for a case of tracking up to three large vehicles. Test case B assumes all targets have random SNRs from 0 to 20 dB, which is a much harder case. Test case C is similar to test case B, but the targets are restricted to the range 13 to 20 dB SNR. This third test case assumes that 13 dB SNR is required to reliably detect a target. In all cases, the SNR in dB is calculated as $10log_{10}$ of the ratio of signal power to the noise power at the input channel of the receiver array.

Table 3.1

Number of Sources Estimation Network Experimental training and testing scenarios

Case	**Training**	**Testing**
A	0: 20,000 1: 20,000 @ 10dB 2: 20,000 @ 10dB 3: 20,000 @ 10dB Total: 80,000	0: 20,000 1: 20,000 @ 10dB 2: 20,000 @ 10dB 3: 20,000 @ 10dB Total: 80,000
B	0: 20,000 1: 20,000 @ rand[0,20] dB 2: 20,000 @ rand[0,20] dB 3: 20,000 @ rand[0,20] dB Total: 80,000	0: 20,000 1: 20,000 @ rand[0,20] dB 2: 20,000 @ rand[0,20] dB 3: 20,000 @ rand[0,20] dB Total: 80,000
C	0: 20,000 1: 20,000 @ rand[13,20] dB 2: 20,000 @ rand[13,20] dB 3: 20,000 @ rand[13,20] dB Total: 80,000	0: 20,000 1: 20,000 @ rand[13,20] dB 2: 20,000 @ rand[13,20] dB 3: 20,000 @ rand[13,20] dB Total: 80,000

3.5 Results and Discussion

Herein, as discussed above, synthesized radar returns from the same range bin are utilized in this study. Returns from different range bins can be processed separately (and similarly for range/Doppler processing). To compare results, the proposed method is compared to AIC, MDL, and EEF using spatial smoothing of the covariance matrix. The proposed method results are shown as confusion matrices and also summarized in terms of the overall accuracy, the percentage of correct entries, underestimates and overestimates. Also, the overall accuracies for the proposed method, AIC, MDL, and EEF are compared.

3.5.1 Training Parameters

Table 3.2 shows the training parameters used for the networks. In all cases, stochastic gradient descent with momentum [7]. The network is updated relative to the classification cross entropy loss assuming multiple mutually exclusive classes.

Table 3.2

Number of Sources Estimation Network Training parameters

Learning Rate	Momentum	Max. Epochs	Mini-batch Size
0.05	0.92	50	500

3.5.2 Test Case Analysis

For the datasets used herein, the training data is independent of the testing data, and the training data is only used to train the network. For all testing, the test data is utilized (the AIC, MDL, and EEF comparison methods do not require training).

A breakdown of results using confusion matrices are provided in tables 3.4, 3.6, and 3.8. These tables can be interpreted as follows. The top row indicates the correct number of sources in a signal, whereas the first column indicates the network output. Therefore, the diagonal terms correspond

Table 3.3

Proposed deep Number of Sources Estimation network architecture
PB = Processing Block: Fully Connected, Parametric ReLU, Batch Normalization,
DR = Dropout, SM = Softmax, CL = Classifier

Layers	Type	Size
	Input	78
1-3	PB	234
4-6	PB	468
7-9	PB	780
10	DR	50%
11-13	PB	4
14	SM	4
15	CL	4

to correct estimates. Additionally, the upper triangle corresponds to underestimates and the lower triangle corresponds to overestimates.

Table 3.4 shows the confusion matrix and table 3.5 shows the overall results for case A, respectively. These tables show the network demonstrated perfect performance for Case A. Tables 3.6 and 3.7 show results for test case B. This case is much more difficult than case A, as the lower bound for SNR is zero and there is a 20 dB SNR range of the signals. Correspondingly, the network demonstrates a performance drop of approximately 5%. Tables 3.8 and 3.9 show results for test case C. These results achieve near perfect classification accuracy due to both the higher SNR sources and the reduced range of SNR values.

For all cases, the training and testing data results are very similar, indicating the network is not overtraining to the training datasets.

Table 3.4

Proposed Method Case A test confusion matrix

	0	1	2	3
0	20,000	0	0	0
1	0	20,000	0	0
2	0	0	20,000	0
3	0	0	0	20,000

Table 3.5

Proposed Method Case A overall results

Case A	Train	Test
Overall Accuracy (%)	100.000	100.000
Underestimated (%)	0.000	0.000
Overestimated(%)	0.000	0.000

Table 3.6

Proposed Method Case B test confusion matrix

	0	1	2	3
0	20,000	0	0	0
1	0	19,960	622	11
2	0	40	18,553	2661
3	0	0	825	17,328

Table 3.7

Proposed Method Case B overall results

Case B	Train	Test
Overall Accuracy (%)	95.218	94.801
Underestimated (%)	3.886	4.118
Overestimated(%)	0.896	1.081

Table 3.10 shows the proposed method compared to the AIC, MDL, EEF methods. In all cases proposed method's performance is over 6% higher than the compared methods in all cases. Additionally, in all of the test cases, AIC, MDL, and EEF have very similar performance.

Table 3.8

Proposed Method Case C test confusion matrix

	0	1	2	3
0	20,000	0	0	0
1	0	20,000	58	0
2	0	0	19,834	339
3	0	0	108	19,661

Table 3.9

Proposed Method Case C overall results

Case 3	Train	Test
Overall Accuracy (%)	99.453	99.369
Underestimated (%)	0.436	0.496
Overestimated(%)	0.111	0.135

Table 3.10

Overall Number of Sources Estimation Network test results shown as accuracy in percent. AIC, MDL, and EEF utilize spatial smoothing (The best results are in **bold**)

	Proposed	AIC	MDL	EEF
A	**100.00**	89.67	89.48	89.47
B	**94.80**	88.22	87.85	87.88
C	**99.37**	91.92	91.73	91.73

Two questions about the network performance are "how will the network perform with only the covariance matrix or eigenvalues as inputs?" and "How does the full covariance matrix compare to the smoothed covariance matrix?" In order to address these questions, the network was modified for the following cases: (1) the full (e.g. non-spatially-smoothed) covariance matrix only (input vector size of 242), (2) full covariance matrix eigenvalues only (input size 11), (3) concatenation of the full covariance matrix and it's eigenvalues (input size 253), (4) smoothed covariance matrix only (input size 72), and (5) the smoothed covariance eigenvalues only (input size 6). Table 3.11 shows results for the network with these different inputs. From the table, the best results are obtained from the input being the concatenation of the spatially smoothed covariance matrix plus it's eigenvalues. The combined smoothed eigenvalue and covariance matrix demonstrate over 5% improvement over all cases not using spatial smoothing. The network shows significant degradation when only provided with eigenvalues regardless of spatial smoothing. All results featuring the covariance matrix feature acceptable results. Interestingly, the addition of the eigenvalues to the covariance matrix degrades the networks performance when comparing the results using the covariance matrix without spatial smoothing.

Table 3.11

Comparison of Number of Sources Estimation Network test results for case B with different input combinations
C,SC = cov., smoothed cov. E,SE = eigenvalues cov. and smoothed cov.
C&E,SC&SE = cov. and eigs., smoothed cov. and eigs. OA = Overall Accuracy,
UE = Underestimated, OE = Overestimated (Best results in **bold**)

Input (Dim)	E (11)	C (242)	C&E (253)	SE (6)	SC (72)	SC&SE (78)
OA %	63.104	88.710	84.790	63.004	89.526	**94.801**
UE %	14.119	8.249	10.397	11.865	7.746	**4.118**
OE %	22.777	3.041	4.813	25.131	2.728	**1.081**

To test the performance of the network relative to SNR, the network trained on Case B was used to evaluate multiple single SNR datasets. Each single SNR dataset contained 1000 signals for each possible number of sources 0 through 3 for a total of 4000 with all sources set at a single SNR value. 31 single SNR datasets were created for each integer SNR value from -5 to 25 dB. The accuracy as well as the number of over and under estimates for each case are plotted against SNR in Figure 3.2.

This plot shows steep declines in performance outside of the training dataset SNR range of 0 to 20 dB for both low and high SNR regimes. Additionally, a slight decline in accuracy can be observed at both ends of the trained SNR range, specifically 0-5 dB and 15-20 dB. A gradually decline of 5% accuracy is observed as the SNR is reduced from 15 to 5 dB. For low SNR values all errors are underestimates as expected. For high SNR targets a trend of overestimation is observed up until 20 dB when underestimates begin to occur. This behavior is likely caused by the limited maximum output value of the network which prevents 3 source cases from ever being overestimated. These observations indicate that when applying this method to actual radar systems, the network should be trained on a range of SNR values larger that what the radar is expected to observe.

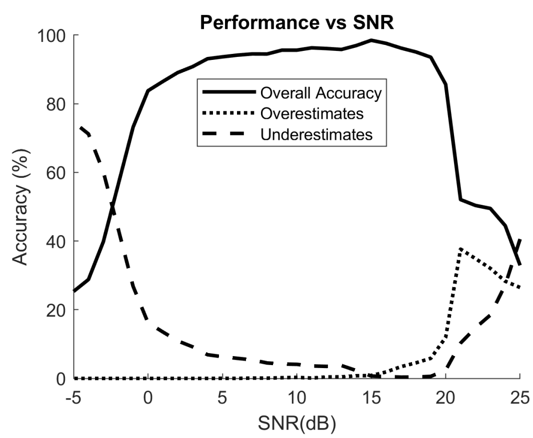

Figure 3.2

Accuracy of the proposed Number of Sources Estimation Network as a function of SNR

The effect of the number of pulses included in each signal was tested by generating datasets with the same parameters as Case B and varying the number of pulses. These datasets correspond to pulse counts of 1-10, 20, and 30. The network was retrained for each of these datasets and

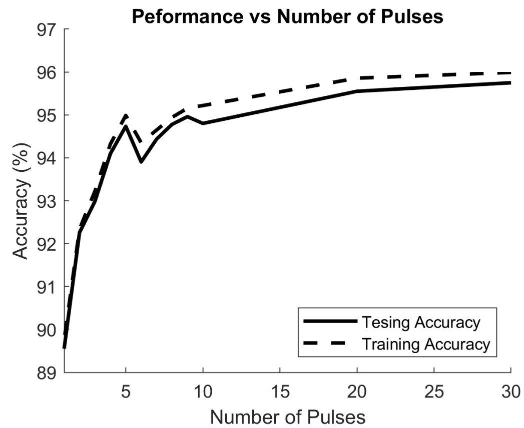

Figure 3.3

Accuracy of the proposed Number of Sources Estimation Network as a function of Number of Pulses

the training and testing accuracy is reported in Figure 3.3. As this plot is constructed from single training instances and the small scale of accuracy variations, the plot is not monotonically increasing but a general trend of increased pulse counts corresponding to increased accuracy is observed. For pulse counts over 10 only 1% improvement is observed despite the increased complexity of computing the signal covariance matrix. When decreasing the number of pulses, the network is able to maintain adequate performance even with a small number of pulses.

Timing experiments were performed to indicate the run time of the network compared to the AIC, MDL, and EEF methods. All experiments were performed using MATLAB 2018B. The PC used for experiments runs Windows Server 2012 with an Intel Xeon E5-2670 CPU, 128GB of RAM, and a Nvidia GTX 970 GPU.

All methods require computation of the smoothed covariance matrix eigenvalues, therefore all methods feature the same pre-processing. This pre-processing time was estimated by computing the smoothed covariance matrix and its eigenvalues for the testing dataset in Case B. This process took 13.7s which corresponds to an average of 171.4μs per signal. The network performance was evaluated in the same manner. The trained network took 3.4s to estimate the number of sources for all signals which corresponds to an average of 42.8μs per signal. The average times for the AIC, MDL, and EEF using the same methodology were 54.5μs, 55.7μs, and 100.8μs respectively. A breakdown of the number of mathematical operations for a similar network is presented in [55].

3.6 Conclusions

Standard solutions for estimating the number of sources, such as AIC and MDL, all fail in cases where the signals present are coherent. However, spatial smoothing is an effective method to address coherent signals.

The proposed DL system which fuses the spatially smoothed covariance matrix and eigenvalues was found to accurately estimate the number of sources, even when the number of receiver channels is small and the number of pulses is also modest. The proposed method outperformed two state–of–the–art methods, namely AIC and MDL, when they also used spatial smoothing for inputs. This is an important contribution, because signal–subspace methods such as MUSIC and MLE require *a priori* estimates of the number of sources. Also, the proposed method does not require matrix inversion or diagonal loading which artificially inflates the noise floor.

Using the spatially smoothed covariance matrix and eigenvalues provides the best results, and the DL network is smaller than the DL network required when the non–smoothed covariance matrix and eigenvalues are used. The downside of any method using spatial smoothing is that there are fewer overall elements, and thus the number of sources that are detectable is smaller than if the full covariance matrix is used. For realistic subarray sizes, this may not be a large issue in practice. Also, spatial smoothing allows a fewer number of receiver elements to be processed simultaneously, which limits the array resolution, as pointed out by [62].

There could be potential electronic countermeasure (ECM) applications, such as detecting how many low power radars are operating in an area. Classification of jamming environments also require estimation of number of sources to make inference on the number of jamming signals affecting each range bin within the radar range swath [10]. There is also potential for non–radar applications, such as modifying the method to not only estimate the number of sources, but also to estimate the relative SNRs of the different sources. This would require adding additional NN regression modules. This approach could have many applications in wireless communications,such as estimating the number of radios talking simultaneously on a channel.

CHAPTER IV

COVARIANCE MATRIX ESTIMATION NETWORK

4.1 Introduction

This chapter presents an analysis of four CNN architectures for Covariance Matrix Estimation and proposes and optimized network. This performance of this network is assessed relative to array construction imperfections. This entire body of work in this chapter has been submitted for publication and includes work intially presented in the following conference paper:

- J. T. Rogers, J. E. Ball, and A. C. Gurbuz, "Data-Driven Covariance Estimation," *2022 IEEE International Symposium on Phased Array Systems Technology (PAST)*, 2022, pp. 1–5.

4.2 Background

The Sample Auto-Covariance Matrix is a common input to many DOA and Number of Sources Estimation algorithms. These algorithms are designed to exploit the desirable characteristics of ideal covariance matrix discussed in section 2.3.1, thus, the performance of these algorithms can be improved by utilizing a more accurate estimation for the auto-covariance matrix.

4.2.1 Conventional Covariance Matrix Estimation

Conventional covariance matrix estimation techniques such as Kang et al's Rank-Constrained Maximum Likelihood Estimation [35, 36], Aubry et al.'s geometric approach [4], and others [80, 40, 29] attempt to adjust the eigenvalues of the sample covariance matrix to conform to the previously discussed values. Therefore, these techniques require a prior estimate of the number of signal sources. As the eigenvalues only correspond to the amplitude information of the signal projected across the array, these techniques have little effect on DoA estimation accuracy. Additionally, these techniques have no effect on subspace DoA estimation algorithms which rely solely on the eigenvectors.

4.2.2 Array Imperfections

Imperfections in the physical construction of antennas and related hardware can create additional sources of errors that affect the quality of the received signal. These problems can be worse for antenna arrays due to variances between antenna elements. Differences in material purity and physical dimensions can cause slight differences in the effective operating frequency of antenna elements. Likewise, variances in the performance of amplifiers attached to these elements can cause distortions in both amplitude and DC offset, as well as introduce inconsistency between the I and Q channels. Finally, for the previously discussed applications of DOA Estimation, variations in the spacing between antenna elements will distort the relative phases of the signal projected across the array according to the relative locations of the signal source and antenna array.

4.3 Methodology

Herein, multiple CNN topologies and configurations are trained and tested to estimate the covariance matrices of signals from a simulated Uniform Linear Array (ULA). The best performing architecture was further optimized and its performance was assessed relative to SNR and the number of pulses used to construct the input sample covariance matrix. Additionally, array imperfections were simulated to assess the networks tolerance to these imperfections Figure 4.1 shows the proposed system with the best performing network architecture.The networks and tests proposed herein is associated is associated with the work presented in [57].

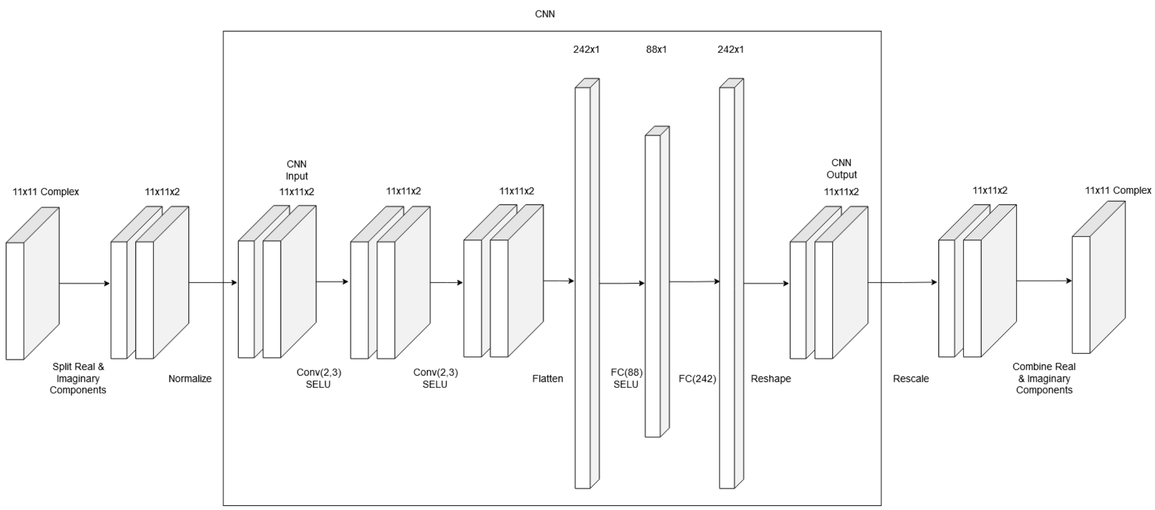

Figure 4.1

Proposed Covariance Matrix Estimation Architecture

4.3.1 Simulated Array Imperfections

To introduce the effects of array imperfections four error terms were introduced into the signal model presented in eq. (4.1) to produce the following signal model:

$$x_m = \left[\sum_{k=1}^{T} (a_k \varepsilon_A + 1) \cdot \exp\left(-j2\pi \frac{f \varepsilon_f + 1}{c} \Delta x\, m \sin(\phi_k) + \varepsilon_\theta \right) \right] + n_m + \varepsilon_{DC} \tag{4.1}$$

The terms ε_A, ε_f, ε_θ, and ε_{DC} correspond to errors in the amplitude, frequency, phase, and DC offset respectively. The amplitude and DC offset errors, in volts, simulate the effects of imperfect amplifier hardware whereas the frequency and phase error correspond to imperfections in the antenna array. The phase errors correspond to alignment imperfections whereas frequency errors are introduced by imperfections in the individual receive elements.

These errors are drawn from zero-mean i.i.d. Gaussian distributions. Multiple distributions with differing variances are used, and the performance of the proposed network is assessed relative to the standard deviation denoted as σ with subscripts matching the corresponding error term.

4.3.2 Simulation Data

Using the signal model presented in 3.3.1 data was generated that simulates an 11 element ULA RADAR operating at 5GHz and 32 snapshots. The receive array element spacing is set to one-half of the wavelength. Signal sources are randomly distributed within a 180° Field of View (FOV) with a minimum spacing of 20°. The simulated signals contain 0 to 4 signal sources with per source SNR values ranging from $-10dB$ to $20dB$ measured at the receiver. The training dataset contains $10,000$ signals for each number of sources providing a total of $50,000$ signals. Likewise, the testing dataset contains $1,000$ signals for each number of sources. Additional testing datasets are used for assessing performance relative to SNR and the number of snapshots. The dataset for the former contains 100 signals for each number of sources and each integer SNR value between $-10dB$ and $20dB$ resulting in a total of $15,500$ signals. The range of snapshots tested includes the first 8 powers of 2, ranging from 1 to 128. 1000 signals are generated for each value of the number of sources and snapshots resulting in $40,000$ signals.

Within these datasets the signals are processed into input-label pairs. The inputs are the sample covariance matrices of the signals computed according to equation 3.2. Likewise, the labels are the true covariance matrix computed according to equation 3.4. To mitigate the effect of the SNR range the network inputs and labels are normalized. As the norm of the true covariance matrix would not be available in any practical application, the Frobenius norm of the sample covariance matrix is used for all normalization. As shown in Figure 4.2, the input sample covariance matrix is normalized using its Frobenius norm for each input. The same factor is used to normalize the true covariance matrix, which serves as the data label. Finally this factor is multiplied by the network output to return it to the appropriate scale. The normalized form of the input and labels are pre-computed before training the networks.

When assessing the tolerance to array imperfections the same training data was used, the same. However, additional evaluation datasets were generated for each of the simulated array imperfections. These datasets contain 100 samples for each potential number of sources value for each level of imperfection. The imperfection levels are determined by the variance of the distribution from which the errors are generated as discussed in section 4.3.1. Ten values logarithmically spaced from 0 to the maximum variance are used as the levels of imperfections. For the array alignment (phase) imperfections the maximum is $\frac{\pi}{2}$ radians. The maximum for amplitude and frequency imperfections is 2. The DC offset maximum is 5. For I-Q variance, the same variances for the previous four imperfections are simulated simultaneously, but independently for the real and imaginary components of the signal.

As support for complex-value neural networks is not yet widespread, the real and imaginary components of these matrices are separated to form two $[M \times P \times 2]$ multidimensional arrays.

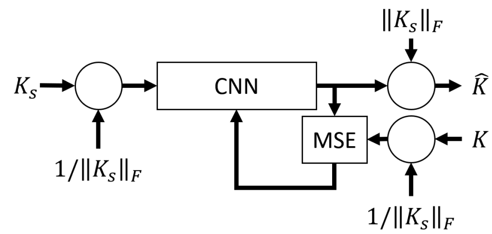

Figure 4.2

Normalization Flowchart

4.3.3 Utilized Neural Network Structures

The evaluated network architectures are all Convolutional Neural Networks (CNNs) utilizing of two dimensional convolution layers. These layers can vary in number of convolutional masks, mask size, and mask stride. Additionally, a convolutional layer can employ zero-padding. Convolutional transpose layers also utilize a set of masks with the same parameters to expand the input size by multiplying all values of the mask by a single sample from the input.

Fully Connected layers are also utilized. These layers consists of an arrangement of multiple linear perceptrons. These layers can vary in the number of perceptrons used.

The Scaled Exponential Linear Unit (SELU) [38] is the sole activation function used in these networks. The SELU is defined in equation (4.2) where λ and α are predefined constants set to 1.05070098 and 1.67326324 respectively. This activation function is self-normalizing eliminating the need for batch normalization layers. Additionally, the SELU, unlike the more commonly used Rectified Linear Unit (ReLU), can return negative values which are necessitated by the Hermitian nature of the estimated matrices. A SELU function follows all convolutional layers and fully connect layers in the evaluated networks except for the output layers.

$$SELU(x) = \lambda \begin{cases} x & x \geq 0 \\ \alpha e^{x} - \alpha & x < 0 \end{cases}, \tag{4.2}$$

All networks are trained using batch processing. The Mean Square Error (MSE) with a learning rate of 0.001 is used as the loss function. All networks are trained for 100 epochs with a batch size of 100.

4.3.4 Error Metrics

Each network topology is assessed using the Normalized Mean Square Error (NMSE). Because the scale of the covariance matrix is influenced by the signal power, noise power, and number of signal sources, normalized error metrics are required. The NMSE utilizes the Frobenius Norm of the true covariance matrix to normalize the error as shown in (4.3) where K is the true covariance matrix, $\hat{K}$ is an estimated covariance matrix, and $|| \cdot ||_F$ is the Frobenius norm.

$$NMSE[\hat{\mathbf{K}}] = \frac{||\hat{\mathbf{K}} - \mathbf{K}||_F}{||\mathbf{K}||_F} \tag{4.3}$$

4.4 Results and Discussion

All network's performance were assessed by taking the average NMSE for 5 training instances. Outlying error values for individual networks were observed, but only for larger-than-average errors which indicates inconsistent training convergence. These outliers were not observed for results presented in this section.

4.4.1 Evaluated Architectures

Four major networks architectures were evaluated. The first network utilized two convolutional layers followed by two matching convolutional transpose layers to produce a symmetric structure which constricts the size of the intermediate data structures. The second network uses four convolutional layers with zero-padding to maintain the same size for first two dimensions for each intermediate data structure. The third network expands the size of the intermediate data structures by utilize two convolutional transpose layers, followed by two symmetric convolutional layers. The final network uses a traditional CNN structure with two convolutional layers and terminates with two fully connected layers.

Table 4.1 shows the layers and relevant parameters for each network. Layers are indicated using the following notation: Convolutional layers are denoted as Conv(Number of Masks, Mask Size), Convolutional Transpose as ConvT(Number of Masks, Mask Size), Fully Connected as FC(Number of Perceptrons), Batch Normalization as BN, and Scaled Linear Unit as SELU. The layer parameters presented in table 4.1 were selected to allow a fair comparison between the 4 proposed architectures.

The results for the four tested architectures are given in table 4.3 as well as the performance using the sample covariance matrix. All four networks demonstrated improved performance compared to the sample covariance matrix with network 4 demonstrating the best overall performance.

As the fourth network contained significantly more learned parameters than the other three, a second experiment was performed after balancing the number of learned parameters. The modified parameters include the number of neurons in perceptron layers and the number of masks in the convolutional and convolutional transpose layers. The balanced networks are shown in table 4.2.

Table 4.1

Covariance Matrix Estimation Network Architectures

Network 1	**Network 2**	**Network 3**	**Network 4**
Conv(32,5)	Conv(32,5)	ConvT(32,3)	Conv(32,5)
SELU	SELU	SELU	SELU
Conv(32,3)	Conv(32,5)	ConvT(32,5)	Conv(32,3)
SELU	SELU	SELU	SELU
ConvT(32,3)	Conv(32,3)	Conv(32,5)	FC(242)
SELU	SELU	SELU	SELU
ConvT(2,5)	Conv(2,3)	Conv(2,3)	FC(242)
Number of Learned Parameters			
21,922	37,282	52,642	264,140

Table 4.2

Balanced Covariance Matrix Estimation Network Architectures

Network 1	**Network 2**	**Network 3**	**Network 4**
Conv(64,5)	Conv(48,5)	ConvT(40,3)	Conv(64,5)
SELU	SELU	SELU	SELU
Conv(64,3)	Conv(48,5)	ConvT(40,5)	Conv(64,3)
SELU	SELU	SELU	SELU
ConvT(64,3)	Conv(48,3)	Conv(40,5)	FC(22)
SELU	SELU	SELU	SELU
ConvT(2,5)	Conv(2,3)	Conv(2,3)	FC(242)
Number of Learned Parameters			
80,706	82,034	81,802	81,280

The results for the adjusted networks are shown in table 4.4. Network 4 still exhibits the best performance and was selected for further improvement.

4.4.2 Final Network Results

The final network architecture is shown in Table 4.5. After selecting from the four proposed networks, each parameter was modified and tested one at a time to improve the networks performance. The network utilizes zero-padding as this allows for an increased variety in convolutional

Table 4.3

Covariance Matrix Estimation Network Architectures Results (Best results in **bold**)

	Average NMSE	
	Training	**Testing**
$\mathbf{K}_s$	0.1048	0.1041
Network 1	0.0238	0.0237
Network 2	0.0119	0.0119
Network 3	0.0140	0.0140
Network 4	**0.0060**	**0.0062**

Table 4.4

Balanced Covariance Matrix Estimation Network Architectures Results (Best results in **bold**)

	Average NMSE	
	Training	**Testing**
$\mathbf{K}_s$	0.1048	0.1041
Network 1	0.0167	0.0167
Network 2	0.0108	0.0108
Network 3	0.0120	0.0120
Network 4	**0.0071**	**0.0073**

mask arrangements as well as a decrease in error. The remaining parameters were modified in the following order: convolutional mask size, number of convolutional layers, number of convolutional masks, number of perceptrons in the third layer, number of fully connected layers, batch size, and learning rate.

The final results this network are shown in table 4.6. The final network's error is approximately $1/20th$ of the error when using the sample covariance matrix.

The performance of the network relative to SNR is shown in figure 4.3. The final network demonstrates robust performance relative to SNR as the performance for all tested SNR levels is superior to the sample covariance matrix at high SNR levels.

Table 4.5

Finalized Covariance Matrix Estimation Network Architecture

Conv(2,3)
SELU
Conv(2,3)
SELU
FC(88)
SELU
FC(242)

Table 4.6

Finalized Covariance Matrix Estimation Network Results

	Average NMSE	
	Training	**Testing**
$\mathbf{K}_s$	0.1048	0.1041
Mean	0.0055	0.0056
Minimum	0.0053	0.0053
Maximum	0.0058	0.0059

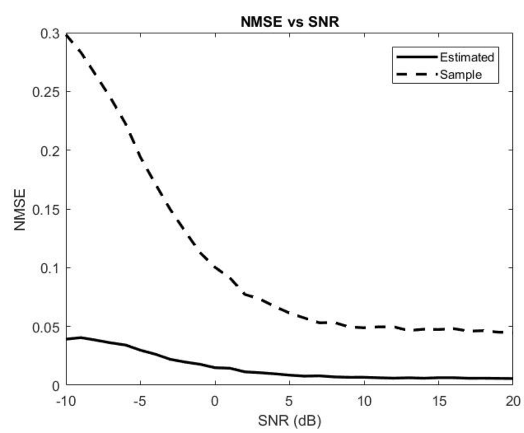

Figure 4.3

Performance vs SNR for Network Results and Sample Covariance Matrix

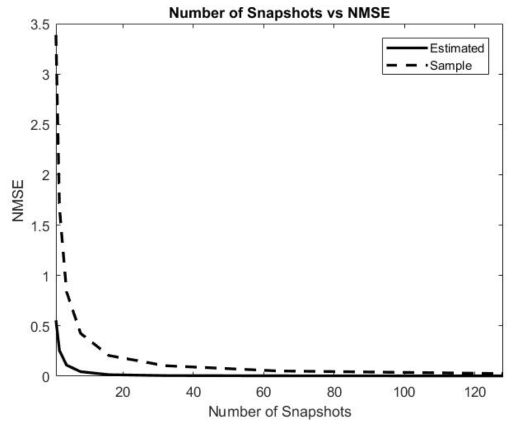

Figure 4.4

Performance vs Number of Snapshots for Network Results and Sample Covariance Matrix

The performance relative to the number of snapshots used when constructing the input sample covariance matrices is shown in figure 4.4 with the exact values given in table 4.7. For all number of snapshots tested, the proposed method demonstrates lower error than the sample covariance matrix. Additionally, the error decreases monotonically as the number of snapshots increases demonstrating the networks ability to generalize despite being trained on a dataset with inputs containing 32 snapshots exclusively.

To demonstrate the benefit of improved estimation of the signal covariance matrix, the MUSIC spectrum of a single target case is shown in figure 4.5. This case was randomly selected from the testing dataset. The target is located at $-23.5°$ with an SNR of $15dB$.

4.4.3 Results with Simulated Array Imperfections

Herein, the behavior of the Covariance Matrix Estimation NMSE relative to the four distorted factors (amplitude, frequency, phase, and offset) is presented.

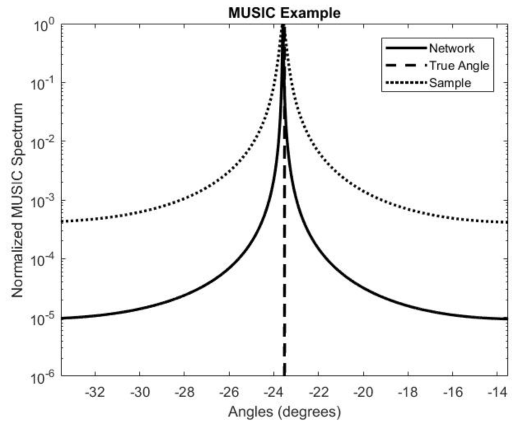

Figure 4.5

Normalized MUSIC Spectrum using the Network Results and Sample Covariance Matrix

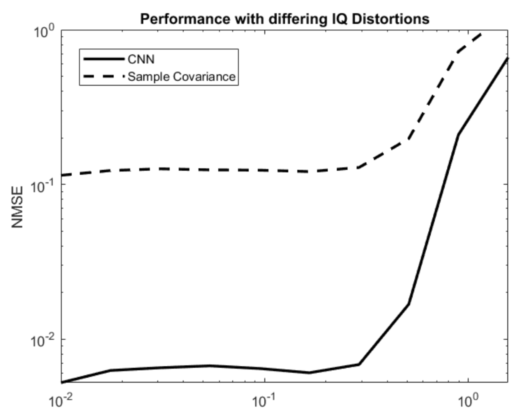

Figure 4.6

Performance vs Phase Distortions

Figure 4.6 show the performance relative to the phase distortions associated with imperfect array element arrangement. The performance maintains satisfactory performance until the phase distortion variance is greater than 0.2, and doesn't surpass the error level of the undistorted Sample Covariance matrix until a variance of at least 0.5.

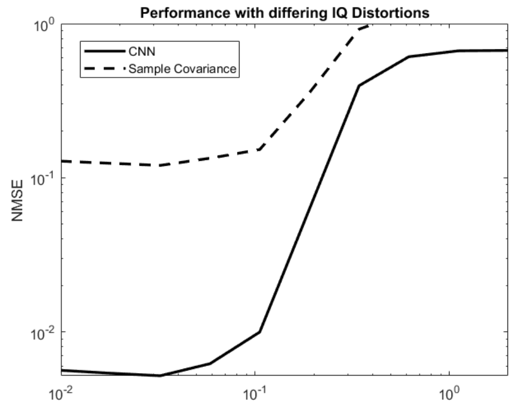

Figure 4.7

Performance vs Frequency Distortions

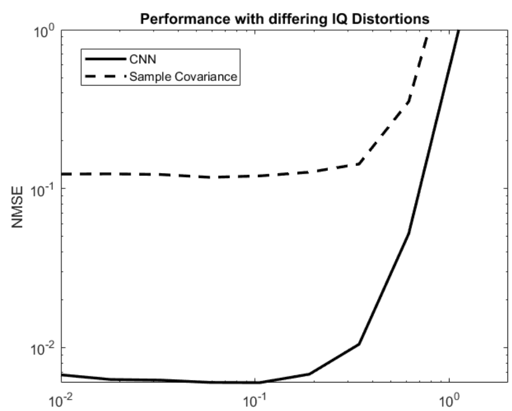

Figure 4.8

Performance vs Amplitude Distortions

The performance relative to frequency distortions, shown in figure 4.7, degrades at lower levels of distortion variance. However, the network still outperforms the undistorted Sample Covariance Matrix with frequency distortions up to approximately 0.1. This is the earliest breakdown of the four distortions examined.

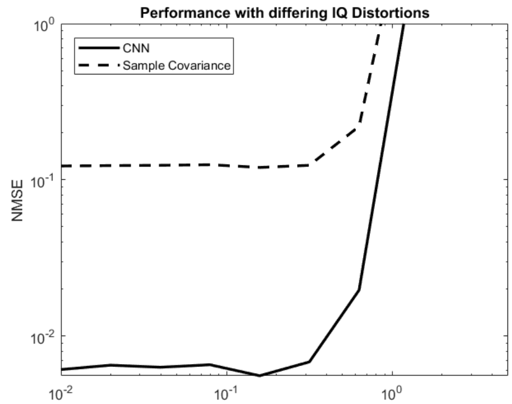

Figure 4.9

Performance vs DC Offset Distortions

The behavior of the performance relative to amplitude, shown in figure 4.8, is similar to the the phase distortions with performance degrading at approximately 0.2.

Figure 4.9 shows the performance relative to DC offsets. The network was most resilient to these distortions which maintain performance until a variance level of at least 0.3.

The performance of all four distortions applied simultaneously with different distortions on the I and Q channels is shown in figure 4.10. As this distortions used different maximum variances in order to utilize appropriate values, multiple axes are given to show the error level for each of the distortions that were simultaneously applied to each step of the dataset. Having different distortions on the I and Q channels appears to have little effect as the performance begins to degrade at approximately the same level as the frequency distortions.

Additionally, for all five distortion scenarios, the proposed network outperforms the sample covariance matrix for the entire evaluated region.

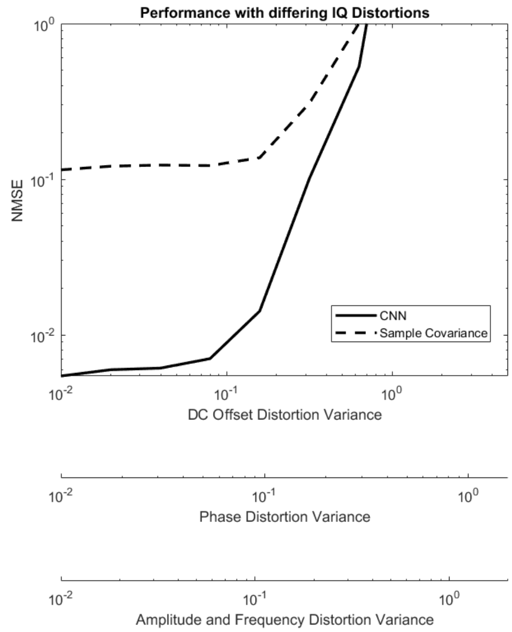

Figure 4.10

Performance with Differing IQ Distortions

4.5 Conclusions

Herein, a comparison of simple neural networks representative of common CNN configurations is presented. The traditional architecture of using convolutional layers followed by fully connected perceptron layers demonstrated the universal best performance. This network was further optimized and demonstrated improved performance compared to the sample covariance matrix optimization. The proposed network exhibited reduced error and significantly improved resilience to lower SNR levels. Additionally, the proposed network can consistently match the performance of the sample covariance matrix with $1/8th$ the amount of snapshots. This proposed network demonstrates improved estimation accuracy of the auto-covariance matrix that is robust to imperfections in the sensor array and related hardware. The network maintains it's accuracy up to error variances of at least 0.1 for all tested perturbations: phase, frequency, amplitude, DC offset, and IQ variations.

These improved covariance matrix estimates can directly benefit DOA algorithms as well as any other array processing algorithm which uses the covariance matrix as an input.

Table 4.7

Results with Varied Number of Snapshots using Proposed Network Estimates and Sample Covariance Matrices

Snapshots	**Estimate**	**Sample**
1	0.5547	3.3872
2	0.2553	1.6747
4	0.1104	0.8388
8	0.0451	0.4254
16	0.0151	0.2057
32	0.0063	0.1045
64	0.0043	0.0524
128	0.0036	0.0262

CHAPTER V

DIRECTION OF ARRIVAL ESTIMATION

5.1 Introduction

This chapter presents a comparison of six CNN architectures for DOA Estimation. The work presented herein has not yet been published or submitted for publication elsewhere.

5.2 Background

The estimation of the Direction of Arrival of the a signal with stationary beams is much more difficult than with scanning sensors. Within this domain, digital beamforming techniques, such as the Minimum Variance Distortionless Response (MVDR) [9], employ steering vectors to manipulate the covariance matrix of a signal. This manipulation facilitates the projection of signal power across an angular spectrum. Subspace algorithms leverage the favorable properties of the eigen-decomposition of the covariance matrix to effectively separate signals from noise, segregating them into orthogonal subspaces. For instance, the MUSIC algorithm [61], employs beamforming on eigenvectors corresponding to the noise subspace. While subspace methods like MUSIC or ESPRIT necessitate knowledge of the number of targets, some techniques like MVDR are capable of generating an angular power spectrum through beamforming alone. All of these techniques are considered super-resolution methods, as they can achieve more precise localization than the Rayleigh Resolution [73]. However, it's important to note that super-resolution techniques, in general, demand extensive computational resources and are often too slow for real-time implementation [17].

The Maximum Likelihood Estimator (MLE) serves as a highly accurate D)A estimation method, but even its efficient implementation [82], can impose significant computational burdens. The MLE requires a prior estimate of the number of sources and, like MUSIC or MVDR, involves a parameter sweep. As the dimensionality increases, the computational complexity of the MLE grows exponentially.

More recently, DoA estimation approaches have emerged based on the spatial sparsity in the angle domain. These approaches utilize techniques like compressive sensing [64, 23], smoothed-ℓ_0 norm [44], sub-Nyquist sampling [46], or quadrilinear decomposition [76] to successfully address DOA estimation problems. Sparsity-based DOA approaches either solve constrained optimization problems or employ greedy methods such as Orthogonal Matching Pursuit (OMP) [70]. However, the selection of constraints or stopping criteria for these techniques typically requires prior knowledge of the number of sources or a reliable estimation of the noise level in the measurements.

5.3 Methodology

Herein, six neural network architectures are constructed and evaluated on two datasets. The architectures shown in figure 5.2 are all CNNs and explore multiple combinations of a novel four layer CNN for DOA paired with the Covariance Matrix Estimation network from Chapter IV. Additionally, multiple transfer-learning style weight initialization schemes are explored.

5.3.1 Network Architecture

Six network architectures (shown in figure 5.2) were designed, trained, and evaluated. These architectures were constructed from a variety of combinations of two component networks: the covariance estimation network from Chapter IV and new DOA Estimation network.

The DOA Estimation network design was adapted from the covariance matrix estimation network. The same composition of layers was used with the size of the final layer adjusted to 181 to output an angular power spectrum with an FOV of $\pm 90°$. Additionally, the network was modified to function as a binary classification problem. To support this, a sigmoid activation function was added to the final layer and the network was trained using a Binary Crossentropy loss function. Initial testing revealed that layer size parameters used for Covariance Matrix Estimation were too small to support DOA. The final proposed DOA estimation network component is shown in figure 5.1.

The architectures constructed from the Covariance Matrix Estimation and DOA Estimation network components are shown in figure 5.2. The first architecture is a classical transfer learning approach where Covariance Matrix Estimation network is modified to support DOA. This network takes the first three layers from the proposed Covariance Matrix Estimation network and combines them with the final layer from the DOA Estimation network. The first three layers' weights are initialized to the values learned in Chapter IV.

The second architecture evaluates the proposed DOA network component in a standalone configuration. This architecture has no pre-trained weight initialization and takes the sample covariance matrix as an input.

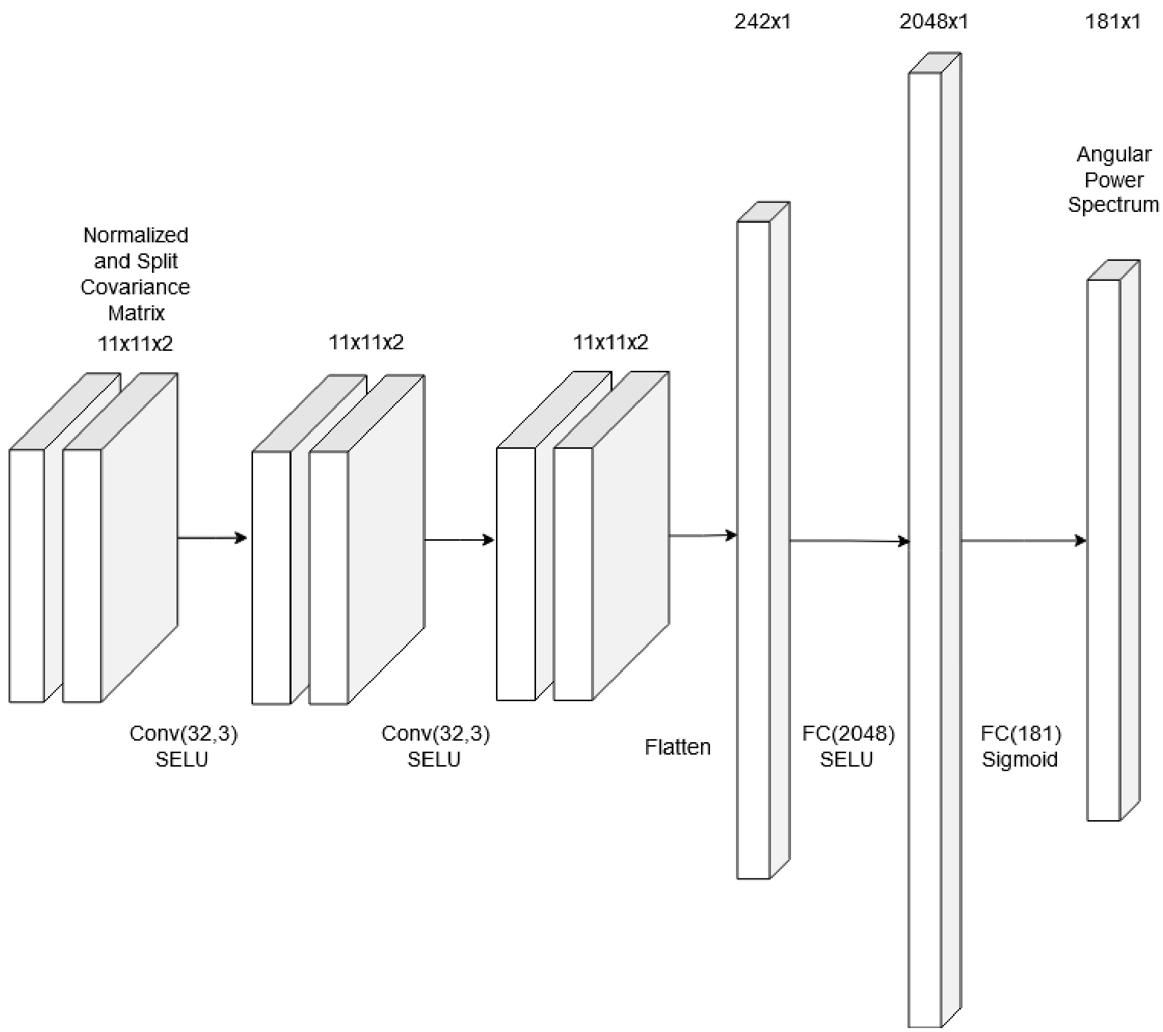

Figure 5.1

DOA Network Component Architecture

The third architecture involves arranging the two network components in sequence. For the Covariance Matrix Estimation the pre-trained network from Chapter IV is used to pre-process all the sample covariance matrices which are then used to train the DOA Estimation network.

The fourth architecture combines the two components as a single network. This combined network is trained on the sample covariance matrices.

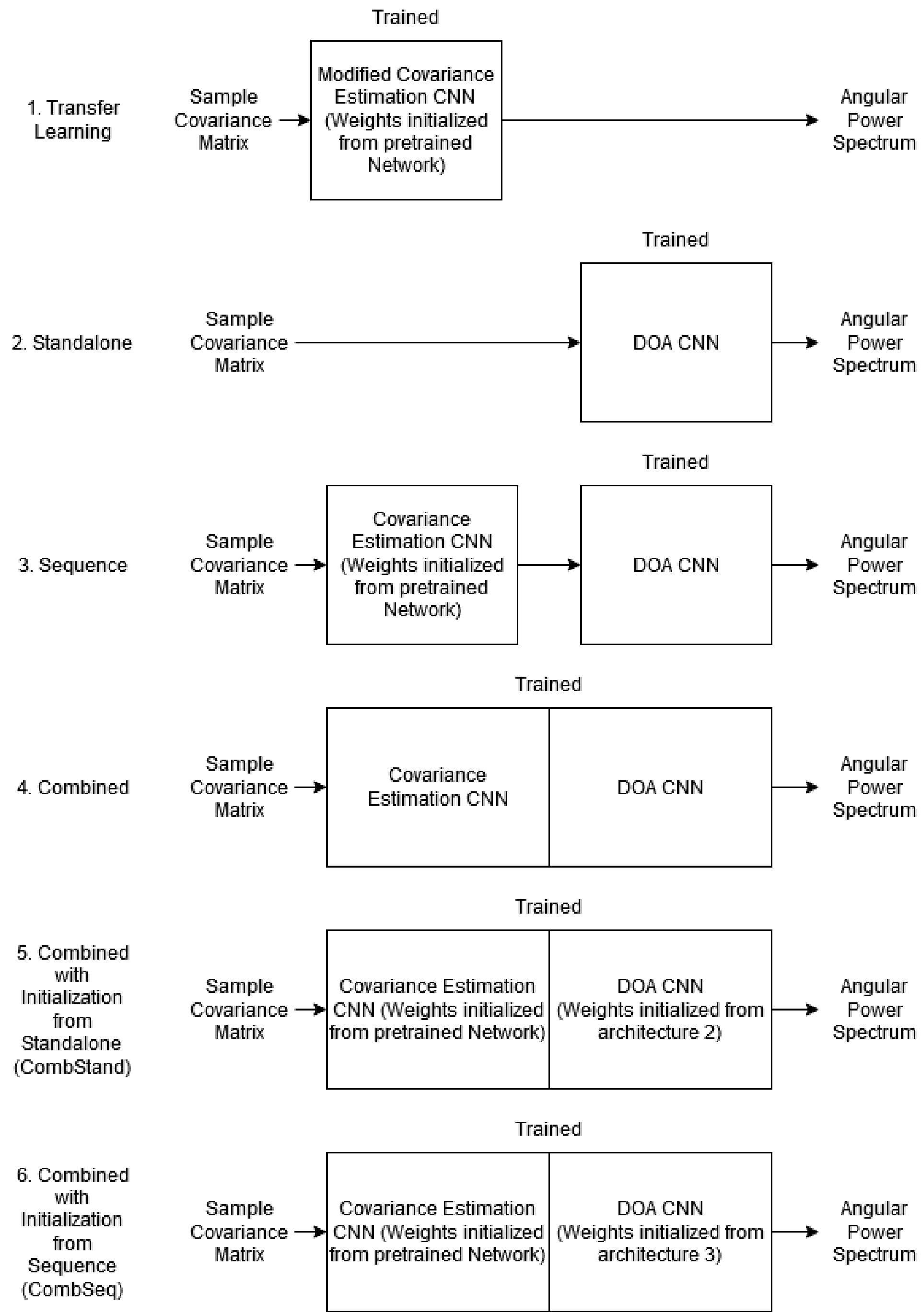

Figure 5.2

Evaluated DOA Network Architectures

The fifth architecture takes the combined networks from the previous architecture, but initializes the Covariance Estimation network component with the pre-trained weights and initializes the DOA Estimation network component with the weights learned in architecture 2.

The sixth and final architecture takes the same approach as the fifth, but uses the weights learned by architecture 3 when initializing the DOA network compoenent's weights.

5.3.2 Simulated Data

The first dataset used herein is a simplified version of the datasets introduced in section 4.3.2. As evaluating DOA accuracy for multiple signal sources involves both the detection and angular estimate accuracy, the simulated signals in this dataset only contained one signal source. Like the previous dataset, the signal simulated an 11-element ULA RADAR operating at 5GHz and 32 snapshots. For initial testing the SNR range was reduced to 0 to 20 dB. The total amount of training data was increased for a total of 100,000 simulated signals with an additional 1,000 for testing.

The second dataset was more challenging. It utilized the extended SNR range, -10 to $20dB$, that was used in the previous chapters. Additionally, the simulated signals could contain up to 4 signal sources. However, unlike the previous chapters, no target cases were not explored as the error metrics used are inadequate to assess the performance for these cases. The final difference with this dataset was the expansion to 1,000,000 training samples.

5.3.3 Error Metrics

Analysis of these architectures was performed using traditional binary classification metrics. The sigmoid activation function bounds the network outputs to the region $[0, 1]$. A threshold of 0.5 is applied to the output to convert it to a binary vector. As only one out of the 181 values for case are expected to be set 1, this problem has a massive class imbalance. For this reason, neither the accuracy nor recall metric are reported. Instead the results are primarily characterized with the probability of detection (P_d) defined as the ratio of accurate signal sources over the total number of signal sources. Additionally the probability of a miss (P_m) defined as the ratio of missed signal sources over the total number signal sources and the probability of a false alarm (P_{fa}) defined as the ratio of the number of false alarms over the total number of samples in the dataset, are reported. A final metric is used to assess the amount of predictions close to accurate. For this metric, the values near all missed signal sources are investigated. If either of the neighboring values in the angular spectrum (corresponding to $\pm 1°$) are 1, the probability of detection is modified as if the case was accurately classified. The modified probability of detection is denoted as $P_d^{\pm 1°}$.

5.4 Results and Discussion

Each of the six architectures were trained 5 times with the average results presented in table 5.1. All architectures gave satisfactory performance with approximately 85% of signal sources estimated as accurately as possible for the resolution used for the angular spectrum.

Table 5.1

DOA Architectures' Results with 1 signal source (Best results in **bold**)

	P_d	P_m	P_{fa}
1. Transfer Learning	0.8514	0.1486	0.1466
2. Standalone	0.8356	0.1644	0.1516
3. Sequence	0.8462	0.1538	0.1510
4. Combined	0.8514	0.1486	0.1566
5. CombStand	**0.8528**	**0.1472**	0.1522
6. CombSeq	0.8360	0.1640	**0.1416**

The worst architecture was the standalone (2) DOA estimation network. This is expected as it only uses one network component and utilizes no weight initialization. Architecture 3, both networks in sequence, has the same concerns, but the improved Covariance Matrix Estimates correspond to a 1% increase in performance.

The combined architecture (4) gives somewhat better performance, but it's P_d is exactly equal to the transfer learning architecture (1) which also has less false alarms. This shows that progressive training schemes can allow accurate results with smaller networks.

The best performing architecture was the combined network using the weight initialization learned in the standalone network (5). Unlike the previous four, this is the first network to utilize the combined structure and weight initialization.

The final architecture's (6) performance was the second worst with almost the same performance as the standalone architecture.

Table 5.2

DOA Architectures' Results with $\pm 1°$ Tolerance (Best results in **bold**)

	$P_d^{\pm 1°}$
1. Transfer Learning	0.9532
2. Standalone	0.9395
3. Sequence	0.9488
4. Combined	0.9600
5. CombStand	**0.9614**
6. CombSeq	0.9398

The results when considering a tolerance of $\pm 1°$ are shown in figure 5.2. All networks demonstrate approximately a 10% increase indicating that approximately two-thirds of false alarms lie within this region. The relative performance of the architecture is mostly unchanged for these results. Architecture 4 now outperforms architecture 1 rather than being tied, while the gap between architectures 2 and 6 narrowed.

Ultimately, these probabilities are too similar to assert that any architecture is clearly superior to another.

Figures 5.3 and 5.4 show scatterplots of the results for the best and worst performing networks, 2 and 5 respectively. Neither architecture produced results with extreme outliers; however, architecture 2 has inaccuracies up to 7° whereas architecture 2 stays within 3°.

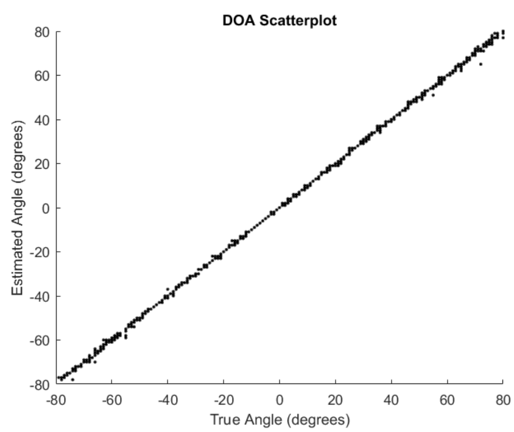

Figure 5.3

DOA Scatterplot with architecture 2. Standalone

5.4.1 Multiple Signal Sources

The same six architectures were also trained on the second dataset described in section 5.3.2. The results for this dataset are summarized in table 5.3. This dataset is much more challenging with poor performance from all architectures; however, the relative performance of the architectures

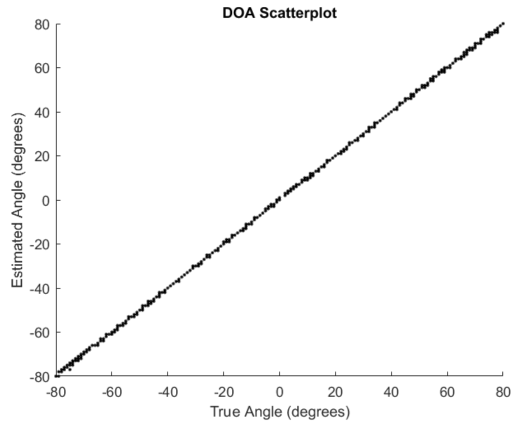

Figure 5.4

DOA Scatterplot with architecture 5. CombStand

differs enough to draw meaningful conclusions. The transfer learning (1) architecture is by far the worst performer with only 40% of signal sources detected at the proper angle. This architecture does have the least amount of false alarms but this is clearly indicative of significant underestimation.

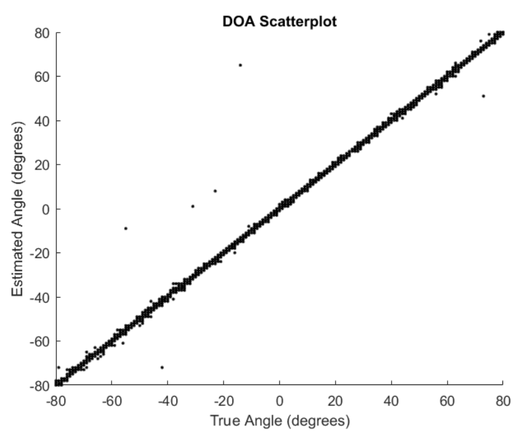

Figure 5.5

DOA Scatterplot with architecture 6, CombSeq, and 1 to 4 signal sources

Table 5.3

DOA Architectures' Results with 1 to 4 signal sources (Best results in **bold**)

	P_d	P_m	P_{fa}	$P_d^{\pm 1^\circ}$
1. Transfer Learning	0.3884	0.6116	**0.1208**	0.4359
2. Standalone	0.5465	0.4535	0.1458	0.6402
3. Sequence	0.4989	0.5011	0.1291	0.5916
4. Combined	0.4779	0.5221	0.1319	0.5559
5. CombStand	0.5475	0.4525	0.1451	0.6485
6. CombSeq	**0.5672**	**0.4328**	0.1700	**0.6728**

The Sequence (3) and Combined (4) architectures demonstrate slightly better performance with P_d's of approximately 50%. Finally, the remaining three architectures all demonstrate similar performance with P_d's of approximately 55%. It is expected that the last two architectures perform best as these are the only architectures to exploit the combined network structure and weight initialization; however, it is surprising that the Standalone (2) architecture achieves similar performance while the Sequence (3) architecture does not as these differ only in the preprocessing of the covariance matrix.

Inspection of the angle scatterplot of architecture six, shown in figure 5.5, shows very few significant outliers with slightly less precision; however, this plot fails to account for missed signal sources. For this plot, $2,588$ signal sources had no corresponding DOA estimate which is approximately one quarter of the $10,000$ signal sources in the testing dataset.

To improve performance alternative thresholds were explored. Inspection of the histograms of the network outputs indicated a significant amount of missed signal sources could be detected with a lower threshold without greatly increasing the number of false alarms. Through trial-and-error a modified threshold of 0.25 was found to achieve significantly improved performance without an

Table 5.4

DOA Architectures' Results with 1 to 4 signal sources and a 0.25 threshold (Best results in **bold**)

	P_d	P_m	P_{fa}	$P_d^{\pm 1^\circ}$
1. Transfer Learning	0.5895	0.4105	0.4869	0.6453
2. Standalone	**0.6852**	**0.3148**	0.3863	0.7936
3. Sequence	0.6460	0.3540	0.3819	0.7461
4. Combined	0.6843	0.3157	0.3860	**0.8167**
5. CombStand	0.6804	0.3196	**0.3714**	0.7934
6. CombSeq	0.6829	0.3171	0.3815	0.8155

excess of false alarms. Additionally, the scatterplots generated with this threshold do not have missed signal sources.

The performance of all architectures with the reduced threshold are shown in table 5.4. The overall performance all architectures improves by approximately 20% but the relative performance of all architectures is mostly unchanged. The transfer learning (1) architecture is still the worst performer, and Sequence (3) architecture is still only somewhat better. However, with the reduced threshold, four architectures (2,4,5,6) achieve the best performance with less than 1% variation in their P_d's.

The scatter plots for all six architectures with the reduced threshold are shown in figures 5.6, 5.7, 5.8, 5.9, 5.10, and 5.11. The number of false alarms included in these scatterplots are summarized in table 5.5. Clearly, these false alarms outnumber the significant outliers in the scatterplots indicated most of the false alarms are multiple detections near a single signal source. All architectures demonstrate good performance with precise angle estimates with a small number of significantly outlying false alarms.

Table 5.5

Number of false alarms in DOA scatterplots

	False Alarms
1. Transfer Learning	746
2. Standalone	572
3. Sequence	514
4. Combined	817
5. CombStand	609
6. CombSeq	620

5.5 Comparison with similar Networks

A similar 8-layer CNN structure is proposed in [50]. This CNN demonstrates accurate performance in the presence of low-SNR signals, however this network was primarily tested on covariance matrices constructed from thousands of snapshots which is too large for many real-time applications. When assessing the performance of the network relative to the number of snapshots, a minimum of 100 snapshots was used which is more than the 32 used for the six architectures presented in this chapter. At 100 snapshots and two targets, the RMSE in degrees for the network presented in [50] was between 1° and 1°. For compassion the RMSE for architecture 6 utilizing a threshold of 0.25 with 1 to 4 signal sources is approximately 1.38° with less than one-third the number of snapshots used.

5.6 Conclusions

The six evaluated architectures explored the benefits of a variety of approaches to implementing Transfer Learning. All six architectures produced accurate angular power spectrums for DOA with up to 4 signal sources. Additionally the architectures are robust to low SNR signal sources. The relative performance of the architectures do not sufficiently differ to indicate a single best

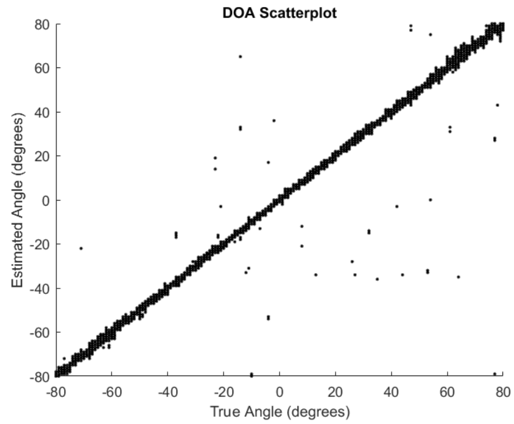

Figure 5.6

DOA Scatterplot with architecture 1, Transfer Learning, with a 0.25 threshold and 1 to 4 signal sources

architecture, but the transfer learning (1) and sequence (3) architectures were clearly inferior to the other four.

The proposed architectures all produce accurate DOA estimates without the need for computationally complex operations such as matrix inversion or eigen-decompostion which are required by conventional algorithms. Additionally, the proposed architectures do not require a prior estimate of the number of signal sources. As the results presented were achieved with simple thresholding, it is clear that the spectrums produced by these architectures do not require a robust peak finding algorithm to produce DOA estimates in the presence of multiple signal sources. Additionally, the proposed architecture demonstrates competitive performance to state of the art techniques [50] when utilizing a practical number of snapshots for estimating the covariance matrix

Future work in this area includes using more training data, testing on more difficult scenarios, and testing with real data. Additonaly, the use of temporal networks could be used to expand this work to angular tracking.

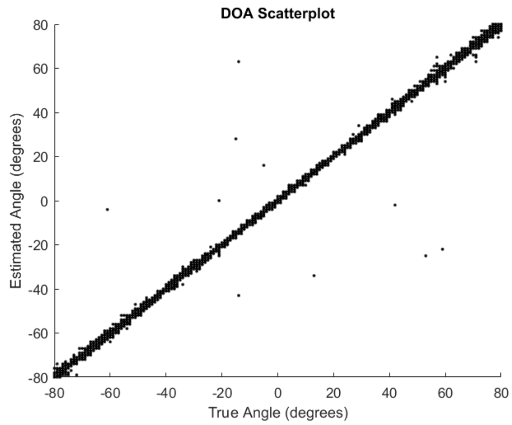

Figure 5.7

DOA Scatterplot with architecture 2, Standalone, with a 0.25 threshold and 1 to 4 signal sources

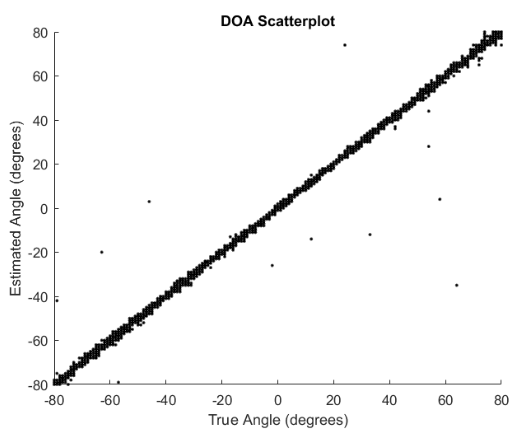

Figure 5.8

DOA Scatterplot with architecture 3, Sequence, with a 0.25 threshold and 1 to 4 signal sources

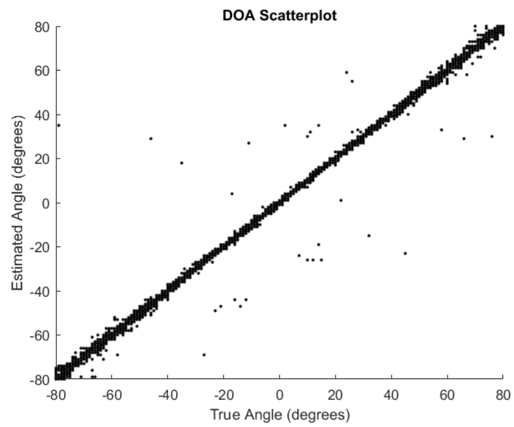

Figure 5.9

DOA Scatterplot with architecture 4, Combined, with a 0.25 threshold and 1 to 4 signal sources

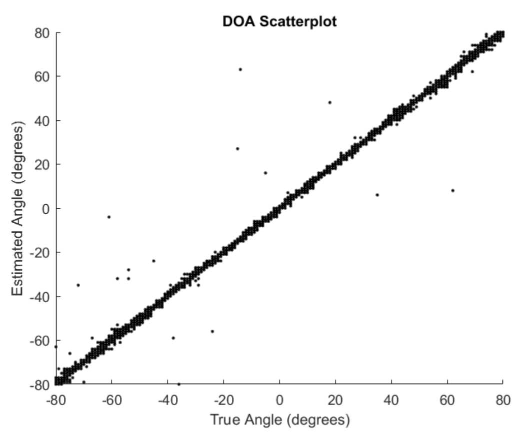

Figure 5.10

DOA Scatterplot with architecture 5, CombStand, with a 0.25 threshold and 1 to 4 signal sources

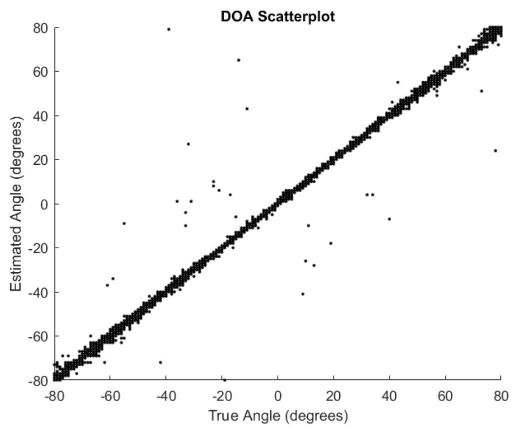

Figure 5.11

DOA Scatterplot with architecture 6, CombSeq, with a 0.25 threshold and 1 to 4 signal sources

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

6.1 Discussion

The network architectures presented herein provide estimates for the related problems of Direction of Arrival (DOA), Number of Sources, and Covariance Matrix Estimation. All the networks produce accurate estimates in the presence of low SNR levels, multiple signal sources, a small array configuration, and constrained number of snapshots. Additionally, the proposed networks are relatively small compared to modern deep networks allowing for efficient implementation.

The proposed Number of Sources Estimation network is more accurate than all three conventional techniques used for comparison. Additionally, the average per-sample speed of the NN implementation was faster than the computation of the conventional techniques for the implementations used herein. Sub-space DOA algorithms can directly benefit from these improved estimates as inaccurate estimates causes improper separation of the noise and signal subs-paces and missed signal sources.

In the area of Covariance Matrix Estimation, four networks were compared and the standard CNN structure provided the most accurate estimates. The optimized variant of this network provides a 20-times more accurate estimate than the Sample Covariance Matrix relative to the Normalized Mean Square Error. Additionally, the proposed network demonstrates robust performance in the presence of array imperfections despite being trained on data assuming a perfect array. As array construction is typically a precise process, the results presented indicate the proposed network should maintain it's performance within the regions of error for an array of acceptable quality. As the Sample Covariance Matrix is used as the network input, the proposed algorithm is more computationally burdensome than the calculation of the Sample Covariance Matrix; however, the proposed network does not require the calculation of eigen-decomposition like other improved Covariance Matrix Estimation algorithms. Additionally, the proposed network estimates the entire matrix rather than only adjusting the eigen-values like the other investigated algorithms [35, 36, 4, 80, 40, 29]. As the information of interest for DOA Estimation primarily lies within the eigen-vectors, these techniques have little benefit for DOA Estimation and have no effect on sub-space algorithms.

For DOA Estimation, six architectures were evaluated. All six architectures demonstrated accurate estimates in the presence of multiple targets and low SNR signal sources. No single architecture demonstrated clearly superior performance with the four best architectures achieving comparable performance in the most difficult scenario tested. The RMSE error in degrees for these architectures is comparable to that of a state-of-the-art network [50] with the proposed networks utilziing less than one-third the number of snapshots.

6.2 Future Work

Future work in this area includes application of these algorithms to real data. As the signal model used herein assumes a narrowband array, this work could also be extended to account for wideband arrays. Additionally, the network architectures could be adapted to accommodate other array geometries.

For Covariance Matrix Estimation, the proposed network should be trained on a specific, real array to determine if the network can learn the array's imperfections for improved performance. Post-processing to enforce known matrix structures, such as the Toeplitz structure in the case of a ULA, could be applied.

Further testing is needed with the proposed DOA architectures to determine the superior architecture. This testing should include a larger training dataset and more difficult scenarios. Additional architectures, which initialize all but the final layer should be constructed to determine the effect of biasing the network towards as specific local minima on the error gradient.

The implementation of complex-valued variants of the networks presented herein should be explored as the data is complex. Deep Residual Networks may also provide improved accuracy; however, these networks would be more computationally complex. Additionally, the benefits of recurrent network architectures, such as transformer networks, could be explored as data in practical applications will be time sequenced. The use of recurrent networks would also allow for the extension to signal source tracking.

REFERENCES

[1] M. Agatonovic, Z. Stankovic, I. Milovanovic, N. Doncov, L. Sit, T. Zwick, and B. Milovanovic, "Efficient neural network approach for 2D DOA estimation based on antenna array measurements," *Progress In Electromagnetics Research*, vol. 137, 2013, pp. 741–758.

[2] H. Akaike, "A new look at the statistical model identification," *IEEE transactions on automatic control*, vol. 19, no. 6, 1974, pp. 716–723.

[3] S.-i. Amari and A. Cichocki, "Adaptive blind signal processing-neural network approaches," *Proceedings of the IEEE*, vol. 86, no. 10, 1998, pp. 2026–2048.

[4] A. Aubry, A. De Maio, and L. Pallotta, "A Geometric Approach to Covariance Matrix Estimation and its Applications to Radar Problems," *IEEE Transactions on Signal Processing*, vol. 66, no. 4, 2018, pp. 907–922.

[5] S. J. Axler, *Linear algebra done right*, vol. 2, Springer, 1997.

[6] A. Barthelme and W. Utschick, "DoA Estimation Using Neural Network-Based Covariance Matrix Reconstruction," *IEEE Signal Processing Letters*, vol. 28, 2021, pp. 783–787.

[7] C. M. Bishop, *Pattern recognition and machine learning (information science and statistics)*, Springer–Verlag, 2006.

[8] Y. Bresler, V. U. Reddy, and T. Kailath, "Optimum Beamforming for Coherent Signal and Interferences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 6, 1988, pp. 833–843.

[9] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, 1969, pp. 1408–1418.

[10] V. Carotenuto, A. De Maio, and S. Iommelli, "Clustering for Jamming Environment Classification," *2020 IEEE Radar Conference (RadarConf20)*. IEEE, 2020, pp. 1–6.

[11] S. Çaylar, K. Leblebicioğlu, and G. Dural, "A new neural network approach to the target tracking problem with smart structure," *Radio science*, vol. 41, no. 5, 2006.

[12] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, 1989, pp. 303–314.

[13] S. Deshmukh and A. Dubey, "Improved Covariance Matrix Estimation With an Application in Portfolio Optimization," *IEEE Signal Processing Letters*, vol. 27, 2020, pp. 985–989.

[14] K.-L. Du, A. Lai, K. Cheng, and M. Swamy, "Neural methods for antenna array signal processing: a review," *Signal Processing*, vol. 82, no. 4, 2002, pp. 547–561.

[15] W. Du and R. L. Kirlin, "Improved spatial smoothing techniques for DOA estimation of coherent signals," *IEEE Transactions on signal processing*, vol. 39, no. 5, 1991, pp. 1208–1210.

[16] A. H. El Zooghby, "Performance of radial-basis function networks for direction of arrival estimation with antenna arrays," *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 11, 1997, pp. 1611–1617.

[17] A. H. El Zooghby, C. G. Christodoulou, and M. Georgiopoulos, "A neural network-based smart antenna for multiple source tracking," *IEEE Transactions on Antennas and Propagation*, vol. 48, no. 5, 2000, pp. 768–776.

[18] A. H. El Zooghby, C. G. Christodoulou, and M. Gerogiopoulos, "Neural Network-based Adaptive Beamforming For One- And Two-dimensional Antenna Arrays," *IEEE Transactions on Antennas and Propagation*, vol. 46, no. 12, 1999, pp. 1997–1999.

[19] A. M. Elbir, S. Mulleti, R. Cohen, R. Fu, and Y. C. Eldar, "Deep-Sparse Array Cognitive Radar," *2019 13th International conference on Sampling Theory and Applications (SampTA)*, 2019, pp. 1–4.

[20] N. J. G. Fonseca, M. Coudyser, J.-J. Laurin, and J.-J. Brault, "On the Design of a Compact Neural Network-Based DOA Estimation System," *IEEE Transactions on Antennas and Propagation*, vol. 58, no. 2, 2010, pp. 357–366.

[21] L. Gan and X. Luo, "Direction-of-arrival estimation for uncorrelated and coherent signals in the presence of multipath propagation," *IET Microwaves, Antennas & Propagation*, vol. 7, no. 9, 2013, pp. 746–753.

[22] E. M. Grais, M. U. Sen, and H. Erdogan, "Deep neural networks for single channel source separation," *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3734–3738.

[23] A. C. Gurbuz, V. Cevher, and J. H. Mcclellan, "Bearing estimation via spatial sparsity using compressive sensing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 2, 2012, pp. 1358–1369.

[24] F. M. Han and X. D. Zhang, "An ESPRIT-like algorithm for coherent DOA estimation," *IEEE Antennas and Wireless Propagation Letters*, vol. 4, no. 1, 2005, pp. 443–446.

[25] H. He, T. Li, T. Yang, and L. He, "Direction of arrival (DOA) estimation algorithm based on the radial basis function neural networks," *Advances in Multimedia, Software Engineering and Computing Vol. 1*, Springer, 2011, pp. 389–394.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[27] A. Hirose, *Complex-valued neural networks: theories and applications*, vol. 5, World Scientific, 2003.

[28] A. Hirose, *Complex-valued neural networks: Advances and applications*, vol. 18, John Wiley & Sons, 2013.

[29] J. Hoffbeck and D. Landgrebe, "Covariance matrix estimation and classification with limited training data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, 1996, pp. 763–767.

[30] H. Hung and M. Kaveh, "Focussing Matrices for Coherent Signal-Subspace Processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 8, 1988, pp. 1272–1281.

[31] H.-S. Hung, "Robust coherent signal-subspace processing for directions-of-arrival estimation of wideband sources," *IEE Proceedings - Radar, Sonar and Navigation*, vol. 141, no. 5, 1994, p. 256.

[32] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[33] S. Jha and T. Durrani, "Direction of arrival estimation using artificial neural networks," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, no. 5, 1991, pp. 1192–1201.

[34] O. d. A. D. Júnior, A. D. D. Neto, and W. da Mata, "Determination of multiple direction of arrival in antennas arrays with radial basis functions," *Neurocomputing*, vol. 70, no. 1, 2006, pp. 55–61.

[35] B. Kang, V. Monga, and M. Rangaswamy, "Rank-Constrained Maximum Likelihood Estimation of Structured Covariance Matrices," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 1, 2014, pp. 501–515.

[36] B. Kang, V. Monga, and M. Rangaswamy, "Computationally efficient toeplitz approximation of structured covariance under a rank constraint," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 1, 2015, pp. 775–785.

[37] Y. Kim and H. Ling, "Direction of arrival estimation of humans with a small sensor array using an artificial neural network," *Progress In Electromagnetics Research B*, vol. 27, 2011, pp. 127–149.

[38] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-Normalizing Neural Networks," *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2017, NIPS'17, p. 972–981, Curran Associates Inc.

[39] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *IEEE signal processing magazine*, vol. 13, no. 4, 1996, pp. 67–94.

[40] O. Ledoit and M. Wolf, "Optimal estimation of a large-dimensional covariance matrix under Stein's loss," *Bernoulli*, vol. 24, no. 4B, 2018, pp. 3791 – 3832.

[41] J. Li and R. T. Compton, "Angle and Polarization Estimation in a Coherent Signal Environment," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 3, 1993, pp. 706–716.

[42] X. Li, A. Alkhateeb, and C. Tepedelenlioğlu, "Generative Adversarial Estimation of Channel Covariance in Vehicular Millimeter Wave Systems," *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, 2018, pp. 1572–1576.

[43] H. Lin, D. Zhou, W. Liu, and J. Bian, "Deep Risk Model: A Deep Learning Solution for Mining Latent Risk Factors to Improve Covariance Matrix Estimation," *Proceedings of the Second ACM International Conference on AI in Finance*, New York, NY, USA, 2022, ICAIF '21, Association for Computing Machinery.

[44] J. Liu, W. Zhou, F. H. Juwono, and D. D. Huang, "Reweighted smoothed l0-norm based DOA estimation for MIMO radar," *Signal Processing*, vol. 137, 2017, pp. 44–51.

[45] K. Liu, K. Ok, W. Vega-Brown, and N. Roy, "Deep Inference for Covariance Estimation: Learning Gaussian Noise Models for State Estimation," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1436–1443.

[46] L. Liu and P. Wei, "Joint DOA and frequency estimation with sub-Nyquist sampling for more sources than sensors," *IET Radar, Sonar & Navigation*, vol. 11, no. 12, 2017, pp. 1798–1801.

[47] Z.-M. Liu, C. Zhang, and P. S. Yu, "Direction-of-Arrival Estimation Based on Deep Neural Networks With Robustness to Array Imperfections," *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 12, 2018, pp. 7315–7327.

[48] T. Lo, H. Leung, and J. Litva, "Radial Basis Function Neural Network for Direction-of-Arrivals Estimation," *IEEE Signal Processing Letters*, vol. 1, no. 2, 1994, pp. 45–47.

[49] G. Ofek, J. Tabrikian, and M. Aladjem, "A modular neural network for direction-of-arrival estimation of two sources," *Neurocomputing*, vol. 74, no. 17, 2011, pp. 3092–3102.

[50] G. K. Papageorgiou, M. Sellathurai, and Y. C. Eldar, "Deep Networks for Direction-of-Arrival Estimation in Low SNR," *IEEE Transactions on Signal Processing*, vol. 69, 2021, pp. 3714–3729.

[51] S. U. Pillai and B. H. Kwon, "Forward/backward spatial smoothing techniques for coherent signal identification," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 1, 1989, pp. 8–15.

[52] C. Qi, Y. Wang, Y. Zhang, and Y. Han, "Spatial difference smoothing for DOA estimation of coherent signals," *IEEE Signal Processing Letters*, vol. 12, no. 11, 2005, pp. 800–802.

[53] E. Radoi and A. Quinquis, "A new method for estimating the number of harmonic components in noise with application in high resolution radar," *EURASIP Journal on Advances in Signal Processing*, vol. 2004, no. 8, 2004, p. 615890.

[54] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, 1978, pp. 465–471.

[55] J. Rogers, J. E. Ball, and A. C. Gurbuz, "Estimating the Number of Sources via Deep Learning," *2019 IEEE Radar Conference (RadarConf)*. IEEE, 2019, pp. 1–5.

[56] J. Rogers, J. E. Ball, and A. C. Gurbuz, "Robust estimation of the number of coherent radar signal sources using deep learning," *IET Radar, Sonar & Navigation*, vol. 15, no. 5, 2021, pp. 431–440.

[57] J. T. Rogers, J. E. Ball, and A. C. Gurbuz, "Data-Driven Covariance Estimation," *2022 IEEE International Symposium on Phased Array Systems & Technology (PAST)*, 2022, pp. 1–5.

[58] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 37, no. 7, 1989, pp. 984–995.

[59] M. Sarevska, B. Milovanovic, and Z. Stankovic, "Alternative signal detection for neural network-based smart antenna," *Neural Network Applications in Electrical Engineering, 2004. NEUREL 2004. 2004 7th Seminar on*. IEEE, 2004, pp. 85–89.

[60] M. Sarevska, B. Milovanovic, and Z. Stankovic, "Reliability of the hidden layer in neural network smart antenna.," *WSEAS Transactions on Circuits and Systems*, vol. 4, no. 8, 2005, pp. 556–563.

[61] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE transactions on antennas and propagation*, vol. 34, no. 3, 1986, pp. 276–280.

[62] T.-j. Shan, M. Wax, and T. Kailath, "On Spatial Smoothing for Direction-of-Arrival Estimation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 4, 1985, pp. 806–811.

[63] T.-J. S. T.-J. Shan and T. Kailath, "Adaptive beamforming for coherent signals and interference," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 3, 1985, pp. 527–536.

[64] Q. Shen, W. Liu, W. Cui, and S. Wu, "Underdetermined DOA estimation under the compressive sensing framework: A review," *IEEE Access*, vol. 4, 2016, pp. 8865–8878.

[65] C.-s. Shieh and C.-t. Lin, "Direction of Arrival Estimation Based on Phase Differences Using Neural Fuzzy Network," *IEEE Transactions on Antennas and Propagation*, vol. 48, no. 7, 2000, pp. 1115–1124.

[66] D. O. Smallwood, "Matrix methods for estimating the coherence functions from estimates of the cross-spectral density matrix," *Shock and Vibration*, vol. 3, no. 4, 1996, pp. 237–246.

[67] M. Solazzi, F. Piazza, and A. Uncini, "Nonlinear blind source separation by spline neural networks," *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*. IEEE, 2001, vol. 5, pp. 2781–2784.

[68] H. Southall, J. A. Simmers, and T. H. O'Donnell, "Direction Finding in Phased Arrays with a Neural Network Beamformer," *IEEE Transactions on Antennas and Propagation*, vol. 43, no. 12, 1995, pp. 1369–1374.

[69] Y. Tan, J. Wang, and J. M. Zurada, "Nonlinear blind source separation using a radial basis function network," *IEEE transactions on neural networks*, vol. 12, no. 1, 2001, pp. 124–134.

[70] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on information theory*, vol. 53, no. 12, 2007, pp. 4655–4666.

[71] S. Unnikrishna Pillai and B. H. Kwon, "Performance Analysis of Music-Type High Resolution Estimators for Direction Finding in Correlated and Coherent Scenes," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 8, 1989, pp. 1176–1189.

[72] K. Upadhya and S. A. Vorobyov, "Covariance Matrix Estimation for Massive MIMO," *IEEE Signal Processing Letters*, vol. 25, no. 4, 2018, pp. 546–550.

[73] H. L. Van Trees, *Optimum array processing: Part IV of detection, estimation and modulation theory*, vol. 1, Wiley Online Library, 2002.

[74] F. Vesperini, P. Vecchiotti, E. Principi, S. Squartini, and F. Piazza, "Localizing Speakers in Multiple Rooms by Using Deep Neural Networks," *Computer Speech & Language*, 2017.

[75] H. Wang and M. Kaveh, "Coherent signal-subspace processing for the detection and estimation of angles of arrival of multiple wide-band sources," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 4, 1985, pp. 823–831.

[76] Z. Wang, C. Cai, F. Wen, and D. Huang, "A quadrilinear decomposition method for direction estimation in bistatic MIMO radar," *IEEE Access*, vol. 6, 2018, pp. 13766–13772.

[77] M. Wax and I. Ziskind, “Detection of the Number of Coherent Signals by the Mdl Principle,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 8, 1989, pp. 1190–1196.

[78] C. Xu and S. Kay, “Source Enumeration via the EEF Criterion,” *Signal Processing Letters, IEEE*, vol. 15, 02 2008, pp. 569 – 572.

[79] W. Xu, F. Gao, J. Zhang, X. Tao, and A. Alkhateeb, “Deep Learning Based Channel Covariance Matrix Estimation With User Location and Scene Images,” 2021, vol. 69, pp. 8145–8158.

[80] Z. Zheng, Y. Zheng, W.-Q. Wang, and H. Zhang, “Covariance Matrix Reconstruction With Interference Steering Vector and Power Estimation for Robust Adaptive Beamforming,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, 2018, pp. 8495–8503.

[81] C. Zhou, F. Haber, and D. L. Jaggard, “A Resolution Measure for the MUSIC Algorithm and Its Application to Plane Wave Arrivals Contaminated by Coherent Interference,” *IEEE Transactions on Signal Processing*, vol. 39, no. 2, 1991, pp. 454–463.

[82] I. Ziskind and M. Wax, “Maximum likelihood estimation via the alternating projection maximization algorithm,” *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP’87*. IEEE, 1987, vol. 12, pp. 2280–2283.