2017-01-01

Classification Of Radar Jammer Fm Signals Using A Neural Network Approach

Ariadna Estefania Mendoza
University of Texas at El Paso, aemendoza@miners.utep.edu

CLASSIFICATION OF RADAR JAMMER FM SIGNALS USING A NEURAL NETWORK APPROACH

ARIADNA MENDOZA

Master's Program in Electrical Engineering

APPROVED:

Benjamin C. Flores, Ph.D.

Berenice Verdin, Ph.D.

Thompson Sarkodie-Gyan, Ph.D.

Melvin Robinson, Ph.D.

Charles Ambler, Ph.D.
Dean of the Graduate School

Dedicación

Dedico esta tesis a mis padres, Manuel Mendoza y Norma Patricia Carrillo, porque sin ellos no estaría aquí y por hacer de mi educación una prioridad sobre todas las cosas. Agradezco mucho sus sacrificios y quiero que sepan que siempre han sido un modelo a seguir en mi vida. Espero se sientan muy orgullosos de mí. Más que nada, les doy las gracias por su amor.

A mi hermana, Myrna y a mi sobrina Paula a las que quiero mucho.

CLASSIFICATION OF RADAR JAMMER FM SIGNALS USING A NEURAL NETWORK APPROACH

by

ARIADNA MENDOZA B.S.E.E.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

THE UNIVERSITY OF TEXAS AT EL PASO

May 2017

Acknowledgements

This research was supported by the Army Research Laboratory (ARL) and by the University of Texas System Louis Strokes Alliance for Minority Participation under grant HRD 1202008. The author gives special thanks to Dr. Berenice Verdin and Mr. James Boehm for reviewing this work and to Dr. Patrick Debroux and his wife for encouraging her to write this thesis.

Most of all the author wants to thank Dr. Flores for being her mentor for the last two years and, also, for his dedication in supporting women in engineering.

Abstract

A Neural Network (NN) used to classify radar signals is proposed for the purpose of military survivability and lethality analysis. The goal of the NN is to correctly differentiate Frequency-Modulated (FM) signals from Additive White Gaussian Noise (AWGN) using limited signal pre-processing. The FM signals used to test the NN approach are the linear or chirp FM and the power-law FM. Preliminary simulations using the moments of the signals in the time and frequency domain yielded better results in the frequency domain, suggesting that time domain training would not be as effective frequency domain training. To test this hypothesis, we developed a training procedure for the NN using either spectra or autocorrelation sequences as inputs as they require a similar amount of signal preprocessing. Classification performance was done in terms of the probability of false alarm (PFA), probability of detection (PD), and probability of error (PE) as a function signal-to-noise-ratio (SNR). In one case, the NN is trained with a set of spectra with either a noisy FM signal with random carrier frequency and bandwidth or strong bandlimited white noise. Simulations show that at an SNR of 5dB, the NN consistently performs signal classification with a PFA close to 0% and a PD higher than 85%. At a SNR of 10dB, the NN reaches a PE of 0%. In another case, the NN is trained with a set of autocorrelations of either a noisy signal or bandlimited noise. At an SNR of 5dB, the NN consistently performs signal classification with a PFA close to 0% and a PD higher than 99%. At a SNR of 10dB, the NN reaches a PE of 0%. In a third case, the NN is trained with a set of signals, which are either linear FM, power-law FM, or bandlimited white noise. Here, at an SNR of 5dB, the NN consistently performs signal classification with a PE close to 0% for both the spectra and the autocorrelation. The conclusion is the NN at a high SNR level

performs exceedingly well for either case. However, at very low SNR, the NN radar signal classifier performs better when its input is the autocorrelation of the signal.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Motivation

In the early morning of December 7, 1941, on the United States Naval Base at Pearl Harbor, Hawaii, a radar operator noticed something peculiar on his radar system. Unknowingly, he was looking at 353 Imperial Japanese fighter planes that were heading to the base. The army lieutenant who led the radar unit at Pearl Harbor dismissed this early warning, thinking it was a false alarm. On this infamous day, our nation was caught by surprise and 2,403 American souls were lost, and 1,178 people were wounded. It is impossible to predict how history would have changed had someone taken heed of the early radar warnings. Since the attack on Pearl Harbor, the country has spent billions of dollars on early-warning radar systems [1], and ever since researchers have been inspired to develop cognitive systems that can assist radar operators and minimize human error. With the development of active countermeasures and counter-counter measures in defense scenarios, the use of cognitive approaches has become ever more appealing.

This thesis is inspired by the events that took place in Pearl Harbor and a task order by the Army Research Laboratory (ARL) which required a study to classify Frequency Modulated (FM) radar jamming signals using a Neural Network (NN) approach. Visualize a passive receiver that mimics the brain [2] and could pick up the enemies' radar signals and correctly classify the signals in a matter of milliseconds. Having such a system could not only provide an effective warning of any impending attacks but also allow the jamming of the enemy's efforts.

Literature Review

We recognize that other methods for classification could be developed using fuzzy logic, but as requested by ARL, we decided to engage in an in-depth study of a NN approach. Others have tested NN against different classification approaches and reported results that are comparable to and, in some instances, superior [3-7]. Such is the case in an experiment by Jordanov *et al.* [8] where three machine-learning models are compared Support Vector Machines (SVM), Random Forest (RF) and NN. Jordanov's test showed that NNs correctly classify radar signals significantly better than the SVM method and only slightly worse than the RF method, thus supporting the use of NNs as a method of signal classification. Ibrahim et al. [7], demonstrated that a multi-layer perceptron NNs perform much better than the K-nearest squares method for forward scattering radar.

Ahalt et al. [3] tested two different kinds of NNs and obtained excellent results for varying Signal-to-Noise Ratio (SNR). In our work, we use Frequency Modulated (FM) signals because of their ability to keep their attributes when corrupted with noise and as in [3] decided to test two different FM signals against varying SNRs. Furthermore, Carter et al. [2] proposed some preprocessing before the radar signals are fed to a NN, such as normalizing the signals and choosing a domain (time or frequency) for analysis. In this light, we decided to use the power spectrum and the autocorrelation of filtered FM signals. Work by Darzikolaei [6] also motivated us to extract the skewness (third moment) and the kurtosis (fourth moment) of the FM signal representations in the time and frequency-domains. The data showed conclusively that there is more variation of these statistics in the frequency domain than in the time domain. Although articles about NN classification uses for radar might be limited, NNs are a popular tool in image processing [11]. For instance Jordanov et al. [8] finds that the performance of NN when trained with missing

sets, when large datasets were used, performed better in a supervised learning environment.

Goals and Methodology

The focus of this thesis is the design of a NN that accepts the power spectrum or the autocorrelation of an FM signal as input. The outcome is the correct classification of signals in the presence of Bandlimited White Gaussian Noise (BWGN). First, the NN is trained and tested using a data set made of signals with imbedded noise. We add signal diversity by considering the following:

a) Linear FM (LFM) corrupted with AWGN
b) Power-law FM (PFM) corrupted with AWGN
c) Bandlimited White Gaussian noise (BWGN).

For hypothesis testing, we consider four scenarioss. In the first scenario, the hypotheses are

H0:The signal is BWGN

H1:The signal is LFM plus AWGN.

In the second scenario, the hypotheses are

H0:The signal is BWGN

H1:The signal is PFM plus AWGN.

In the third scenairo, the hypotheses are

H0:The signal is LFM plus AWGN

H1:The signal is PFM plus AWGN.

Finally, in the fourth scenario, the hypotheses are

H0:The signal is BWGN

H1:The signal is LFM plus AWGN

H2:The signal is PFM plus AWGN.

For scenarios 1, 2, and 3, the objective is to measure the Probability of Detection (PD) and Probability of False Alarm (PFA). For scenario 4, since the NN must select from three options, the objective is to measure false positives (3). In all cases, probabilities are estimated from the NN's confusion matrix.

The above requires further clarification. To begin, the NN requires a training matrix as input. The training matrix is a collection of data sequences derived from the FM signal and BWGN. Tests are conducted with the purpose of determining in which domain, i.e. time or frequency, results are better. Figure 1.1 is a block diagram that shows the basic steps followed when the training matrix is filled with spectrum sequences. Figure 1.2 is as similar block diagram that shows the training matrix being filled with autocorrelation sequences. Notice that in both instances, the process starts by generating noisy FM and BWGN jamming signals, which are immediately smoothed using a Chebyshev filter. The training matrix consists of M spectral sequences, each having N samples. Thus, the training matrix is an MxN matrix. The sample sequences in the training matrix are randomly divided into three groups, 70% for training, 15% for validating, and 15% for testing. Following training and validation, the testing results of the NN are assembled into the confusion matrix from which the PE, PD and PFA are computed. This will be discussed with detail in a subsequent chapter. In this thesis, the software platform of choice is MATLAB®.

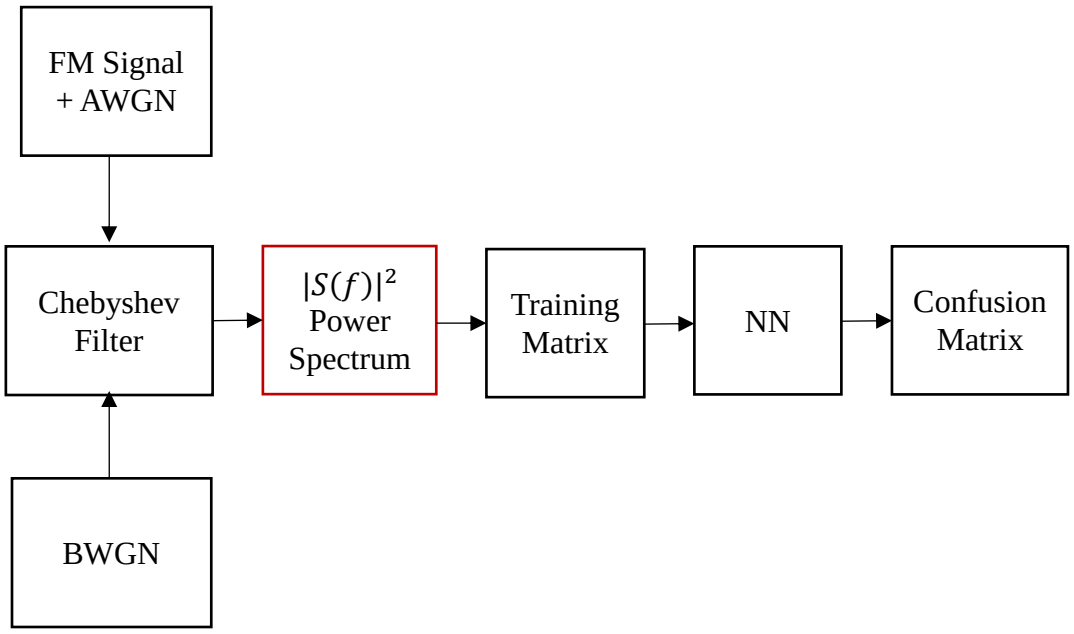


Figure 1.1. Block diagram of neural network classifier in which the training matrix is a set of power spectra.

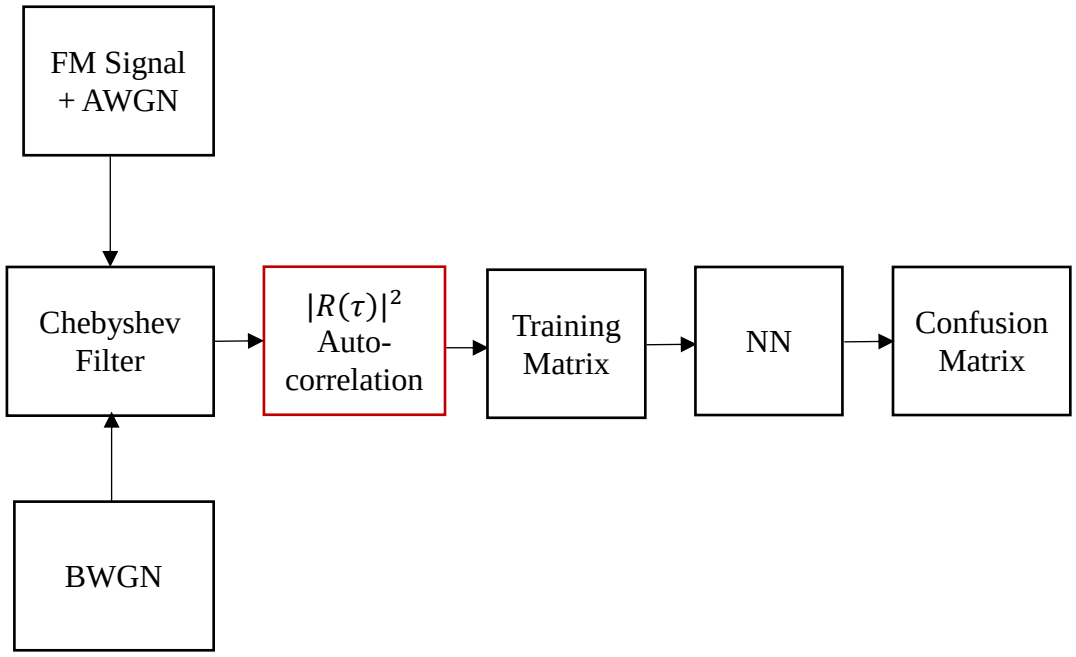


Figure 1.2. Block diagram of neural network classifier in which the training matrix is a set of autocorrelations.

Thesis Overview

This thesis follows a traditional structure for work of this nature. Chapter 2 is a review of Neural Networks and an analysis of the NN attributes used for assessment. Chapter 3 is a detailed description of the FM signals, the way they are pre-processed and how they are fed to the neural network. Chapter 4 describes the tests conducted and corresponding results. Lastly, chapter 5 provides concluding remarks of the work conducted in this thesis.

Chapter 2: Neural Networks

Neural Networks rapidly became a subject of interest in 1943 when the concept of simplified neurons was introduced by McCulloch and Pitts [12], only to subsequently lose popularity because of their limitation to solving linear problems, i.e. problems that could only be solved by drawing a single straight line. NNs gained popularity again in 1974 with the introduction of hidden layers and backpropagation, which overcame previous limitations encountered [13]. Neural networks now have many applications in aerospace, automobile, banking, and medical areas. This research focuses on a defense application although the results could stimulate discussion in work related to target tracking, radar and image signal processing, data compression and image identification [14].

Introduction to Neural Networks

The basic processing unit of a NN is called an artificial neuron, neuron, or node. Figure 2.1 shows a single artificial neuron and a multilayer artificial neural network. In (a), $X_1, X_2, X_{3,} X_4$ are the signal flow inputs and $W_1, W_2, W_{3,} W_4$ are the weights going into the neuron. When the threshold θ is exceeded by the activation function the neuron will fire. The activation function can be a sigmoid, softmax, tan-sigmoid, etc. In (b), the multilayer artificial neural network has three essential parts: the input layer, hidden layer, and the output layer [12]. We chose a multilayer system with one input layer and one hidden layer. Also, we use supervised learning where the targeted output is known a priori and used to feed the NN [12]. Below we discuss how the toolbox works and offer more information on pattern recognition.

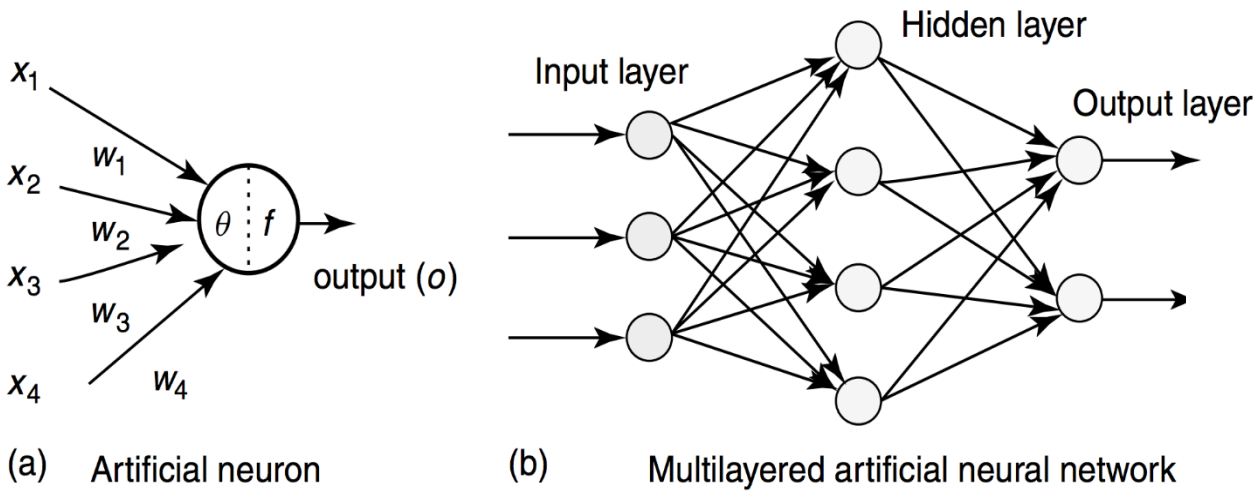


Figure 2.1 Architecture of an artificial neuron and multilayered neural network [12].

Neural networks are made of simple elements working in parallel. These items are made to mimic biological nervous systems. Just like in nature, the connections between elements control the network function [14]. The NN can be trained to perform a function by altering the connections or the weights between the neurons. Neural networks are formed so that an input gives you a specific target output. Figure 2.2 illustrates this situation [15].

Neural Network Toolbox

For this work, we used the NN toolbox in MATLAB® which offer four options: Function Fitting, Pattern Recognition, Data Clustering, and Time-Series Analysis. Furthermore, we used the pattern recognition toolbox with the default settings. More information on how to make a neural network more robust can be found in Soto's work [15].

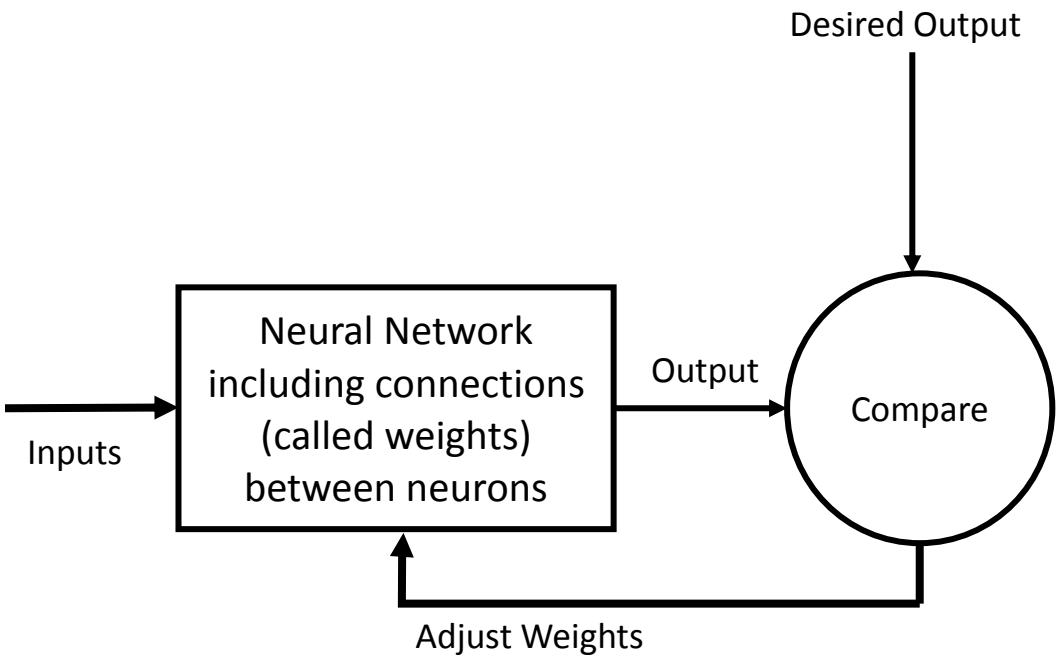


Figure 2.2. Neuron adjusting weights by doing back-propagation [16].

Pattern Recognition

The pattern recognition application is a two-layer feedforward network, with a sigmoid transfer function in the hidden layer,

$$\vartheta(v_k) = \frac{2}{1+e^{-2n}} - 1, \quad (1)$$

where $\vartheta(v_k)$ is the threshold that needs to be exeded for the neuron to fire, and a softmax transfer function in the output layer [16]. The sigmoid function was used because tests done by Soto [15] show that applying a sigmoid transfer function as the activation function to the hidden layer yield better results than any other activation function tested. The default number of hidden neurons is ten neurons [16]. The number of output neurons equals the number of target vectors (output). For the binary hypothesis cases, testing

requires two target vectors. For the three-hypothesis case, testing is done with three target vectors.

To reduce the error at the output, the NN uses the scaled conjugate gradient algorithm. The scale conjugate gradient algorithm is a network training function that updates weight and bias values per the scaled conjugate gradient method [17]. The algorithm can train any network if its weight, net input, and transfer functions have derivative functions. Backpropagation is used to calculate derivatives of performance on the weight and bias variables X. As mentioned above, the NN divides the data into three categories, 70% used for training, 15% for validation which tests if the network is generalizing and it stops training before overfitting, and 15% for testing which is a separate test from training and validation. Training ends when the number of epochs is reached, the maximum time is surpassed, the performance is minimized, or the performance gradient is lower than the minimum gradient.

The network also uses cross-entropy as a method to correct the weights. It works by calculating a vector which contains the network performance given the target and the output. The function heavily penalizes the outputs that are incorrect and penalize the weights that are more accurate very lightly [18]. In the NN architecture, the number of inputs, N, is a variable that changes. When N is large, classification should improve, resulting in a low PFA and high PD.

The neural network is in default mode so as to concentrate on determining which method for preprocessing data (power spectrum or autocorrelation) yields better results. Optimizing the NN is an important topic of research [15]. The following chapter explains in detail how the data was obtained.

Chapter 3: Methodology

Jamming Signals

The first signal of interest is the Linearly Frequency Modulated (LFM) signal. This signal is ideal for this study because of its ability to retain its shape in the frequency domain when noise is added. The time domain representation of the linear chirp is

$$s(t) = Ae^{j2\pi\left(f_0 t+\frac{1}{2}kt^2\right)} \quad 0 \le t \le T, \tag{1}$$

where A is the amplitude, f_o is the initial frequency, and k is the chirp rate. The frequency sweep is $\beta = kT$. Sampling at a rate $f_s = 1/\Delta t$ where t is $n\Delta t$ over the entire pulse yields N samples. Let

$$f_{norm} = \frac{f_0}{f_s} \tag{2}$$

and

$$\beta_{norm} = \frac{\beta}{f_s} \tag{3}$$

so that the discrete-time form of equation (1) becomes

$$s(n) = Ae^{\left(j2\pi f_{norm}n+j\pi\beta_{norm}\frac{n^2}{N-1}\right)} \quad 0 \le n \le N-1. \tag{4}$$

For the purpose of training the NN, both the normalized bandwidth and initial frequency of the signal are uniformly distributed. To guarantee that the signal is wideband and its spectrum is not aliased, the range for the bandwidth is set to

$$0.05 \le \beta_{norm} \le 0.2 \tag{6}$$

while the range for the initial frequency is

$$0.02 \le f_{norm} \le (1-\beta_{norm}). \tag{7}$$

The second signal of interest is the time Power-Law Frequency Modulated (PFM) signal which in continous-time is given by

$$s(t) = Ae^{j2\pi(f_0 t + \frac{kt^{\alpha+1}}{\alpha+1})} \quad 0 \le t \le T \tag{9}$$

where $\alpha \geq 1$ is the power index and k is the chirp rate. The frequency sweep is $\beta = k$. Sampling at a rate $f_s = 1/\Delta t$ over the entire pulse yields N samples. The frequency and bandwidth are normalized as in the case of the LFM. Therefore, the discrete-time version of equation 9 becomes

$$s(n) = Ae^{\left(j2\pi f_{norm} n + j2\pi \frac{\beta_{norm} n^{\propto+1}}{(N-1)^{\propto}(\propto+1)}\right)} \; 0 \le n \le N-1. \tag{10}$$

To train the NN, both the normalized bandwidth and initial frequency of the PFM signal are uniformly distributed using the same ranges specified for the LFM signal.

To conduct an analysis of NN performance versus SNR, Gaussian noise is added to the LFM and PFM signals:

$$r(n) = s(n) + n(t). \tag{11}$$

A set of N noise samples is computed using a zero mean, complex Gaussian random number generator. Recall that the SNR is given by

$$SNR = \frac{A^2}{\sigma^2} \tag{12}$$

where σ is the noise standard deviation which has a range of $0.03 < \sigma < 1$. For simulation purposes, let $A = 1$. Since the SNR is the variable of choice, solving for σ yields

$$\sigma = \sqrt{\frac{1}{SNR}}. \tag{13}$$

Finally, bandlimited white Gaussian noise (BWGN) is utilized in this study as a jammer signal [14]. N samples of jamming noise are generated using a Gaussian random number generator with zero mean and unit variance. Since the spectrum of this signal spans the entire frequency range $0 \le f \le 1$, a chebyshev filter with random center

frequency and bandwidth as in the case of the FM signals is used. Thus the spectrum of the jamming noise is given by

$$\left|N_J(f)\right|^2 = |N(f)|^2 |H_C(f)|^2 \tag{16}$$

where$|N(f)|^2$ is the spectrum of white noise and $H_C(f)$ is the transfer function of the Chebyshev filter. Clearly, the noise is forced to be bandlimited by the bandwidth of the filter. A discussion of Chebyshev filter characteristics follows.

Chebyshev Filter

The Chebyshev filter used in this study achieves a fast roll-off with a minimum ripple in the frequency domain. Figure 3.1 shows a Chebyshev filter gain envelope. Notice the flat response in the bandwidth of interest.

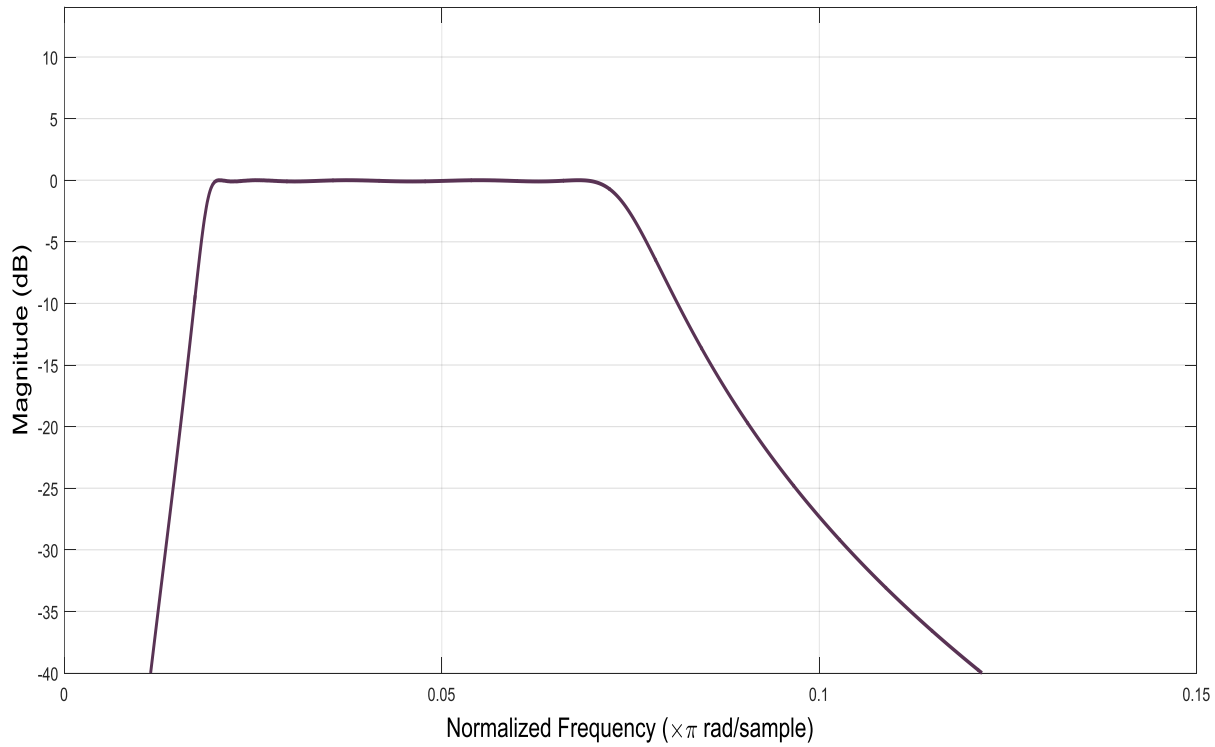


Figure 3.1. Chebyshev magnitude response for randomly generated FM radar jammer signal.

The magnitude-squared function for this filter is [15]

$$|H_c(f)|^2 = \frac{1}{1+\varepsilon^2 V_P^2(\mathrm{f}/\mathrm{f_c})} \tag{17}$$

where $V_P(x)$ is an p^{th}-order polynomial. ε is a user supplied parameter that controls the amount of passband ripple, and $\mathrm{f_c}$ is the upper passband ripple edge. The filter is optimized with respect to the input signal. That is, the routine automatically calculates the filter order and its corner frequencies. The filter requires two additional specifications: peak-to-peak bandpass ripple (set to 0.1dB) and stopband attenuation (set to 40dB).

Power Spectrum and Autocorrelation

The purpose of the power spectrum is to show at which frequencies the power of the jamming signal is concentrated. We calculate the power spectrum by taking the magnitude squared of the discrete Fourier transform. Figure 3.2, 3.3, and 3.4 illustrate the power spectrum for the LFM, PFM, and BWGN. At low SNR, the power spectrum of the FM signals start to look like the power spectrum of the BWGN. For simplicity, we normalize all the signals to their highest peak in an attempt to make the NN classify signals by their spectrum shape and not their absolute power. To avoid spectral aliasing, the highest frequency allowed is 0.5.

The autocorrelation of a signal compares the signal with itself as a function of delay. For wideband FM signals, this is equivalent to compressing the signal with a matched filter so the autocorrelation shows a sharp peak at zero delay. Interestingly, autocorrelation sidelobes that can vary significantly are observed. Figures 3.5, 3.6, and 3.7 show the autocorrelation for the LFM, PFM, and BWGN jamming signals. In all cases, the number of samples N is 1024 and the SNR is 40 dB.

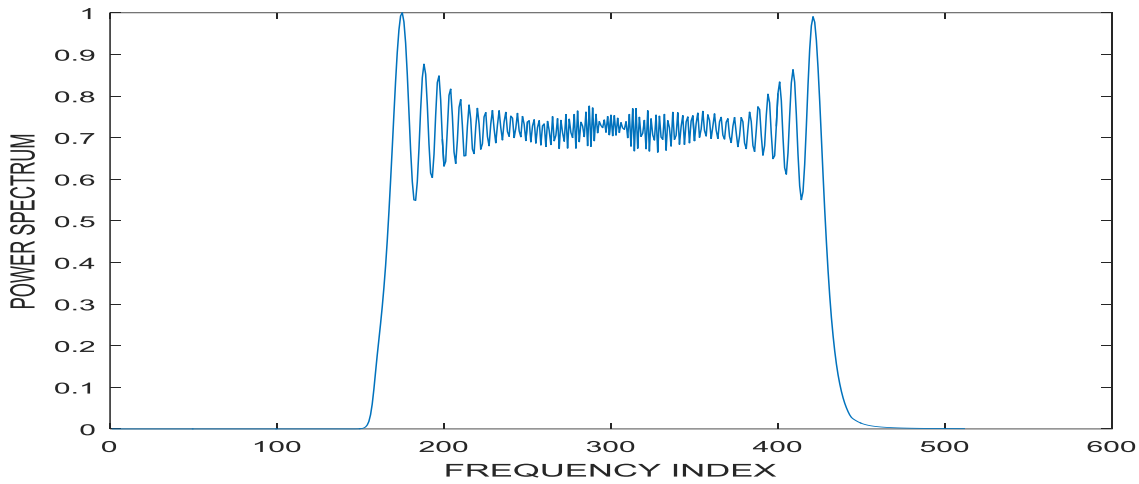


Figure 3.2. Power spectrum for randomly generated LFM

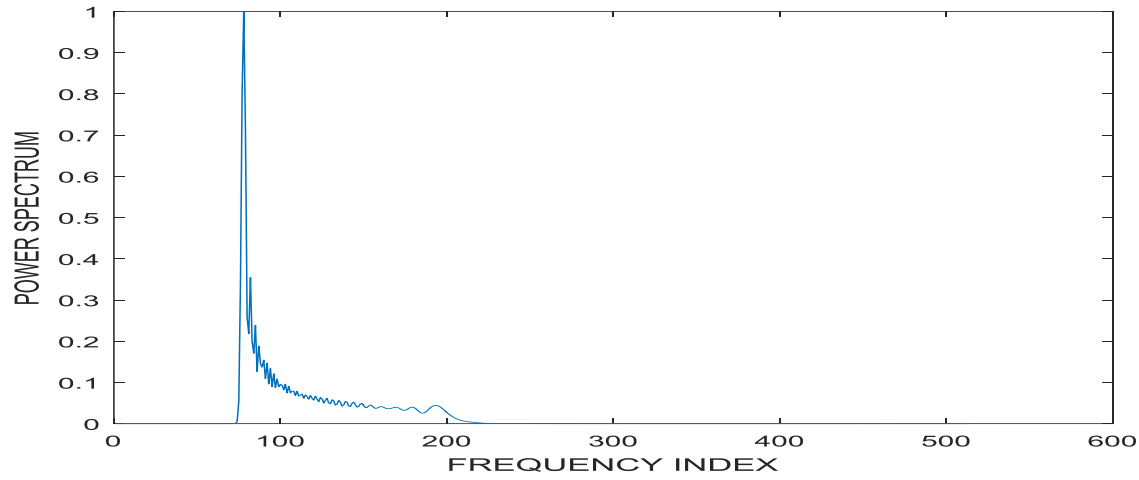


Figure 3.3. Power spectrum for randomly generated PFM

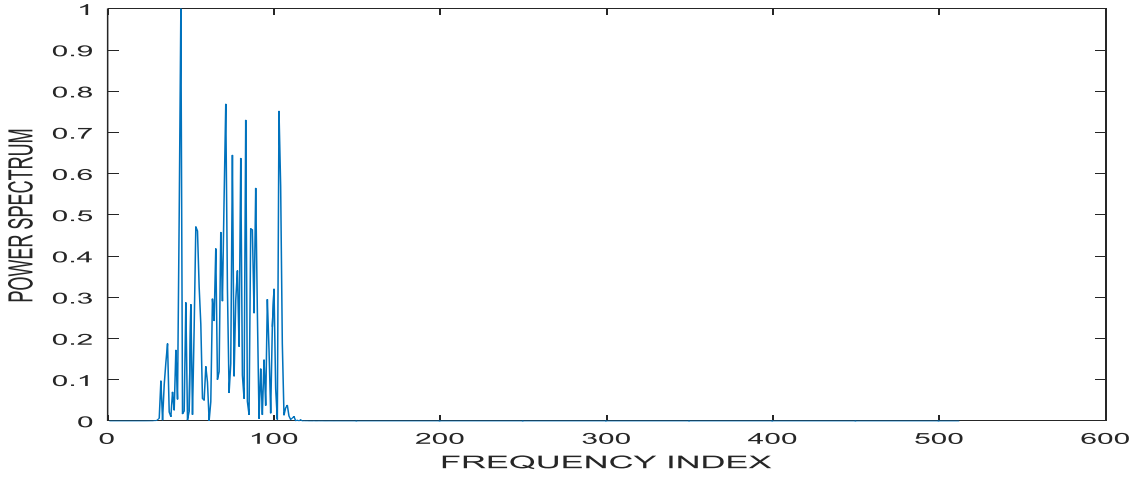


Figure 3.4. Power spectrum for randomly generated AWGN

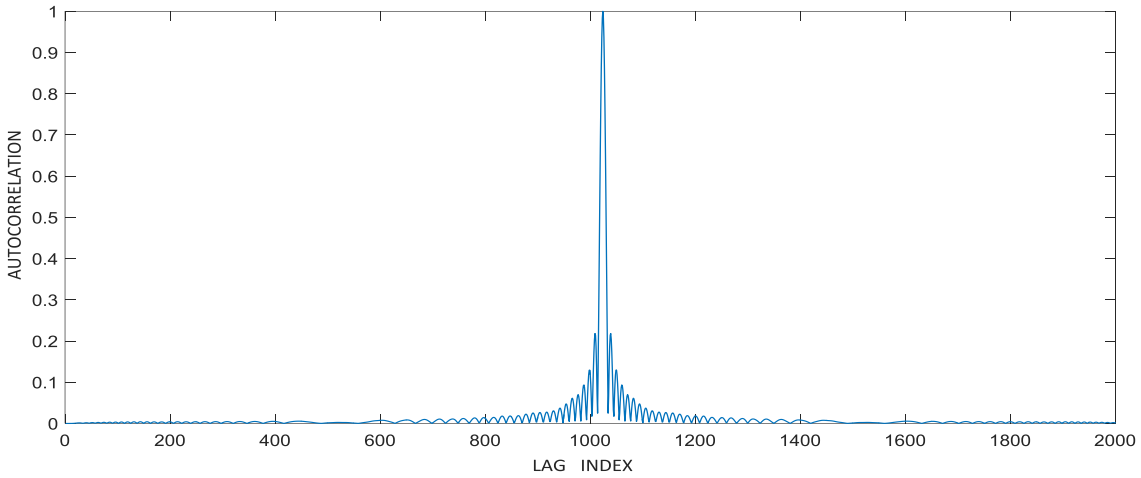


Figure 3.5. Autocorrelation for randomly generated LFM

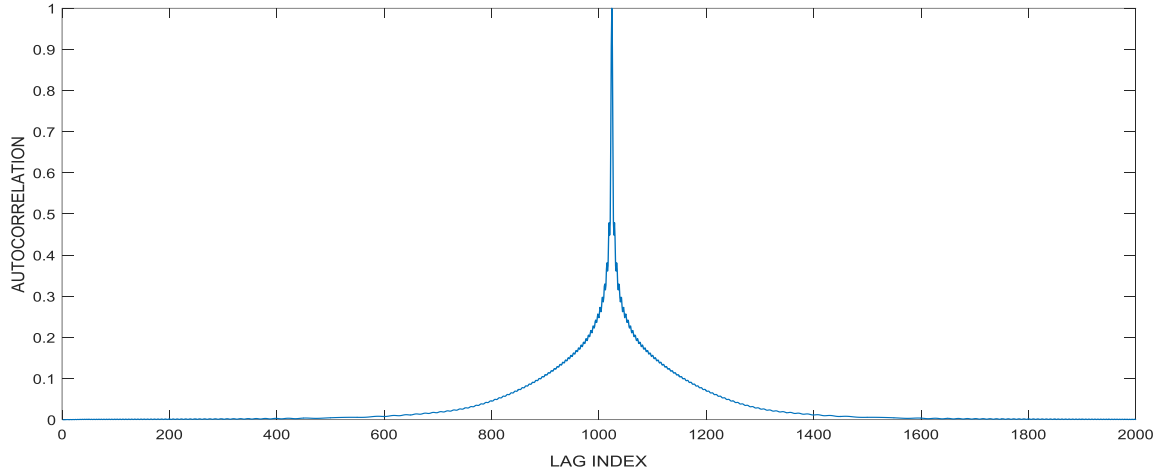


Figure 3.6. Autocorrelation for randomly generated PFM

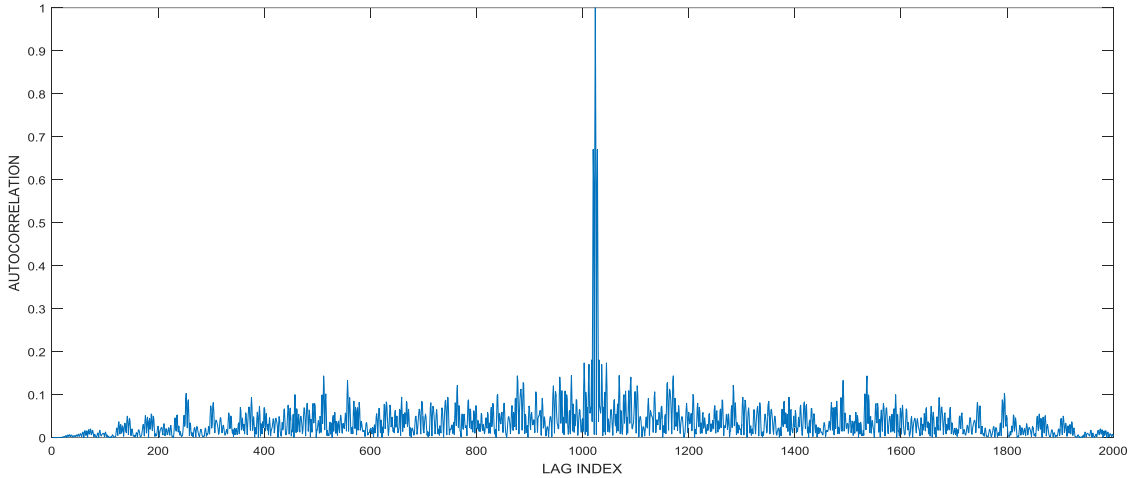


Figure 3.7. Autocorrelation for randomly generated AWGN

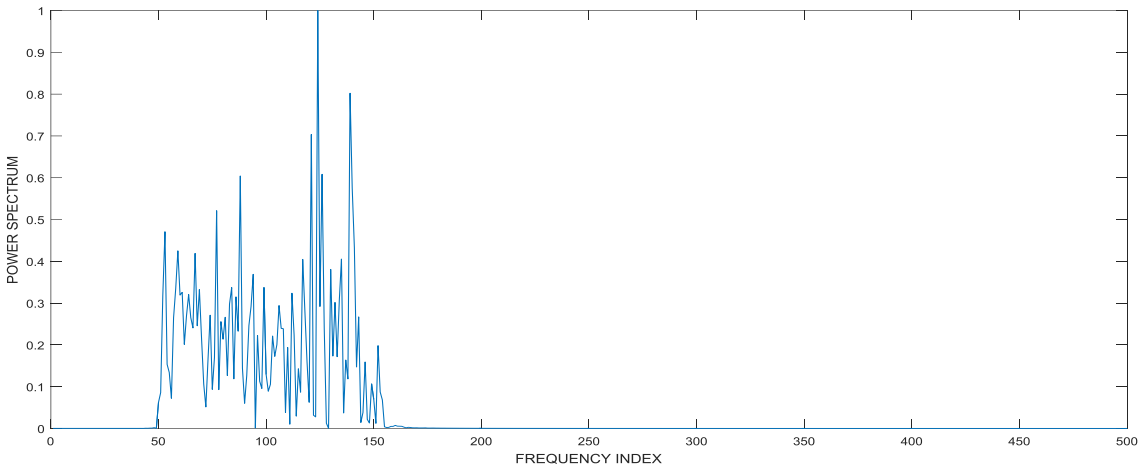


Figure 3.8 Power spectrum for randomly generated LFM at -5 dB SNR

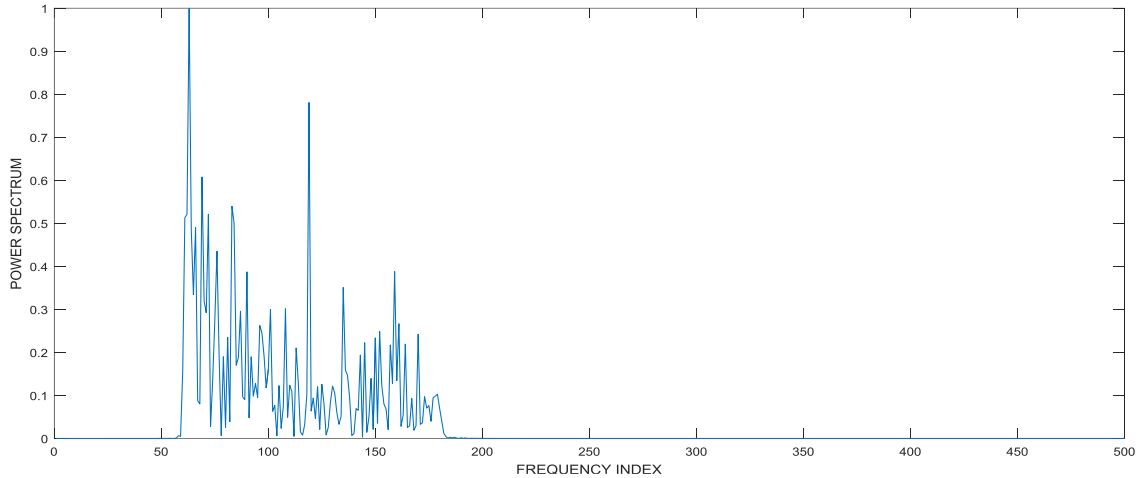


Figure 3.9 Power spectrum for randomly generated PFM at -5 dB SNR

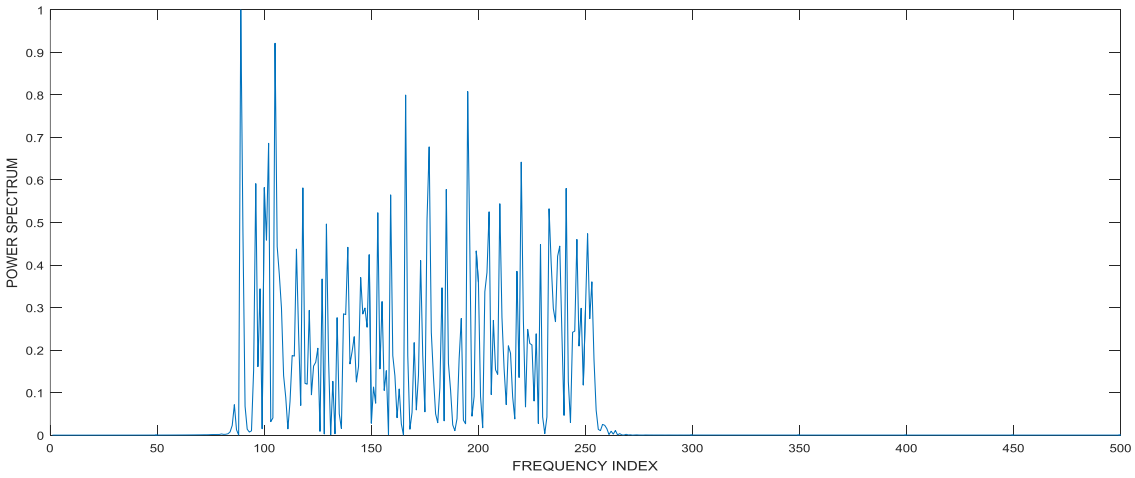


Figure 3.10 Power spectrum for randomly generated BWGN at -5 dB SNR

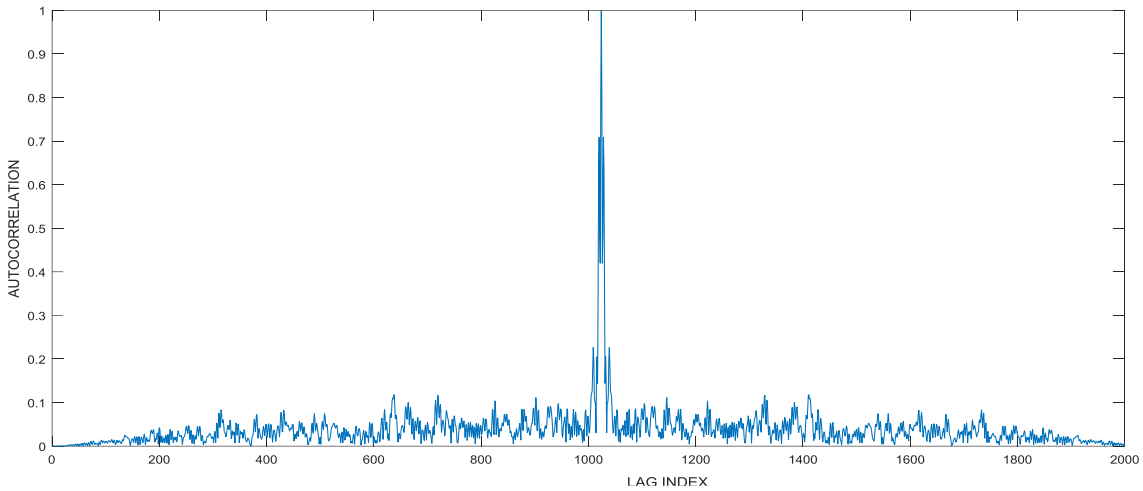


Figure 3.11. Autocorrelation for randomly generated LFM at -5 dB SNR

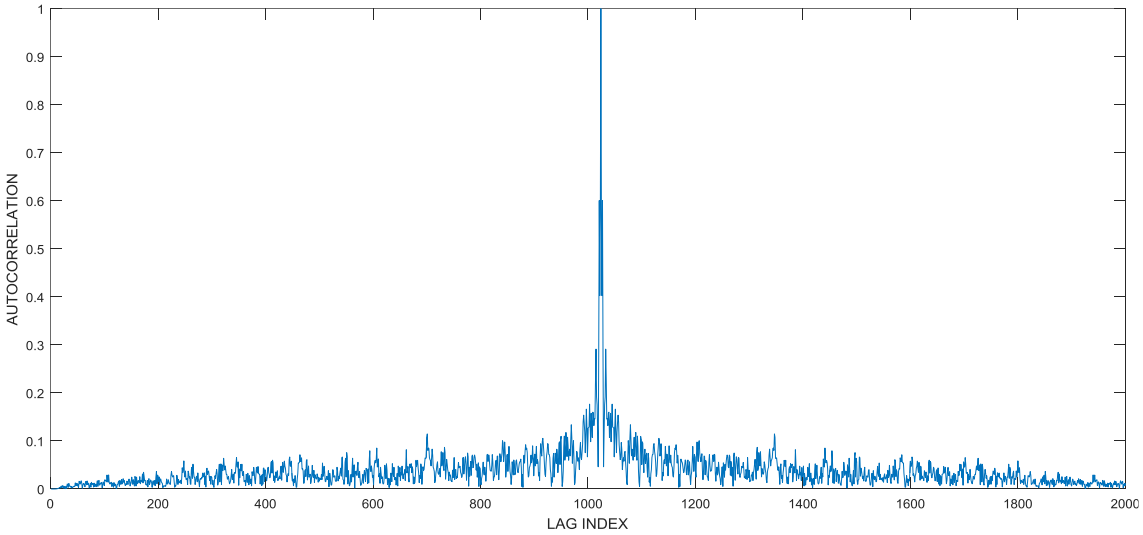


Figure 3.12 Autocorrelation for randomly generated PFM at -5 dB SNR

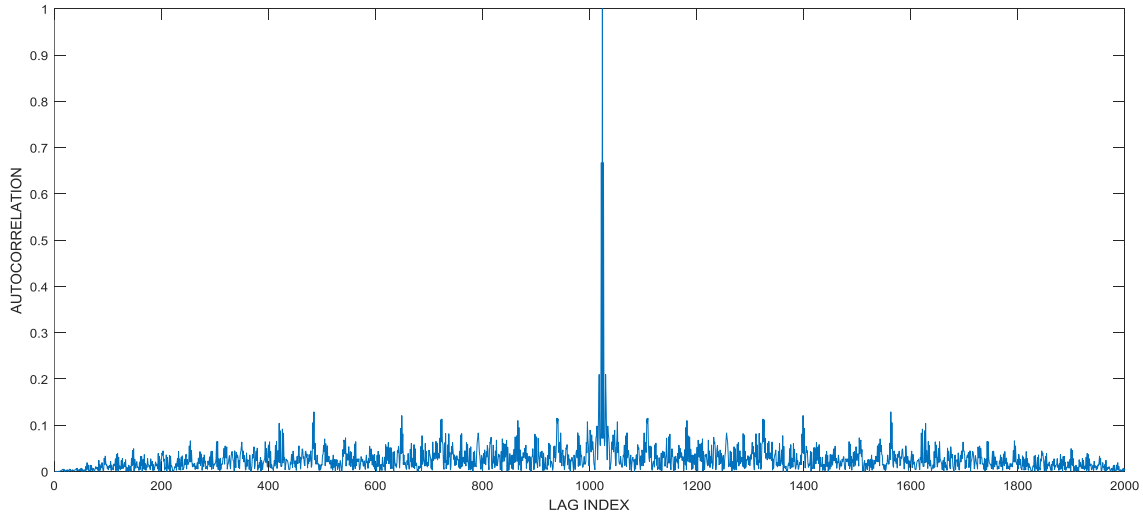


Figure 3.13 Autocorrelation for randomly generated BWGN at -5 dB SNR

Data Preparation

The NN is fed a large number of data for training, validating, and testing. All the required data are stored in a matrix containing either M spectra or M autocorrelations. Each spectrum or autocorrelation has N samples. Figure 3.14 illustrates in (a) how the matrix is saved as when a linear FM and a BWGN are used for input. The first columns are H1 and the consecutive ones correspond to H0. Figure 3.14 (b) illustrates the case where we test the L for the LFM, P for PFM and A for the BWGN as inputs to the NN. Each row is one signal containing N samples and each case contains the same number of signals M.

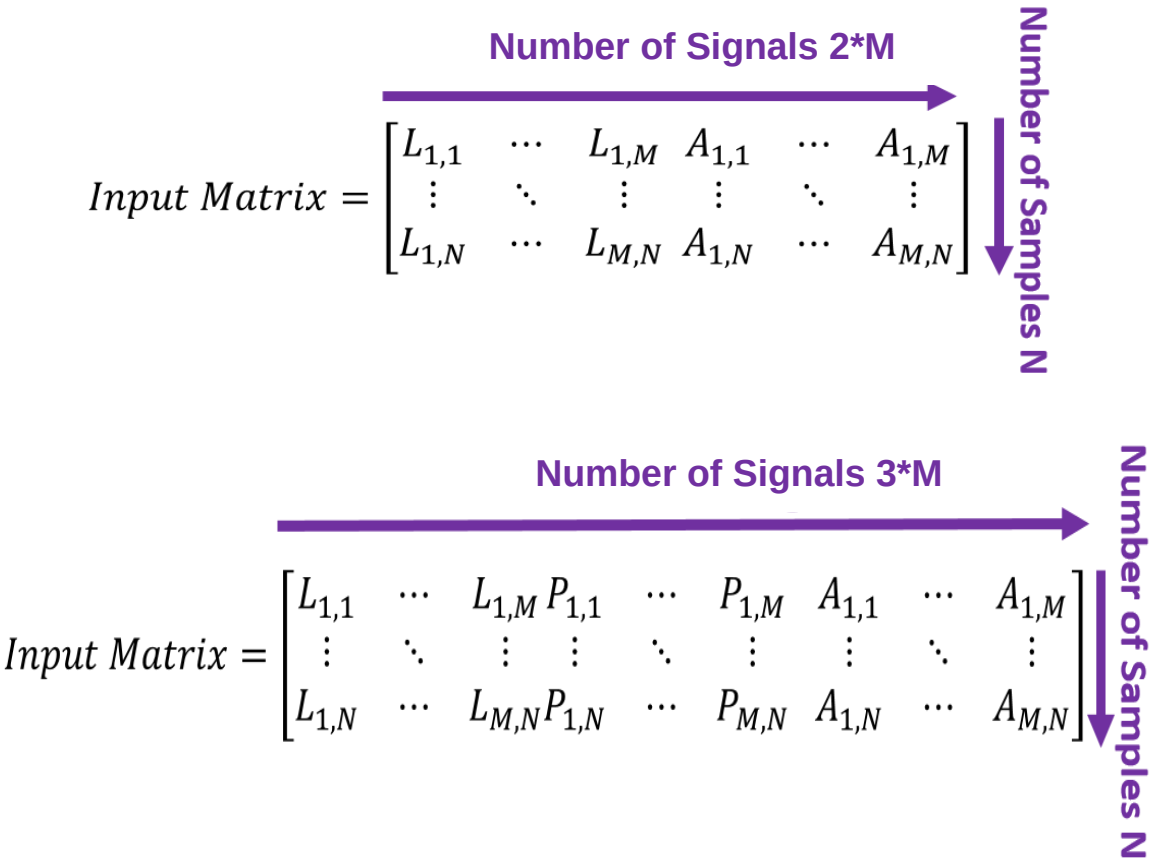


Figure 3.14 (a). Two signals being used for training (b) three signals being used for training

During supervised training, the NN must know which input actually corresponds to which signal. Therefore, the NN fed a target matrix in binary format that represents the actual signals being used as inputs. The number of rows in the target matrix is equal to the number of hypotheses tested and the number of columns equals the same number of possible decision combinations. For binary hypothesis testing, the decision to be made is whether the jamming signal is bandlimited noise or one type of frequency modulation. In this instance, the target matrix is

$$Target\ Matrix = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Here, hypothesis H0 is represented by the first two columns of the matrix while hypothesis H1 is represented by the last two columns. For example, if we were to test LFM signals versus BWGN, the first two columns would represent the LFM and the next two columns would be for BWGN.

When the decision to be made is whether the jamming signal is one of three signals (i.e. LFM, PFM, or BWGN), the target matrix is

$$Target\ Matrix = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

In this case, the first two columns of the matrix represent hypothesis H0 (the signal is BWGN), the two middle columns represent H1 (the signal is LFM), and the last two columns represent hypothesis H2 (the signal is PFM).

Chapter 4: Results

This chapter summarizes the results of numerous tests of the NN signal classifier to quantify its performance. Figure 4.1 shows the way we obtain measurements for the following tests. Here, the Probability of Detection (PD) is estimated as the fraction of times the NN output indicates that the signal is FM when, indeed, the input is an FM signal. In contrast, the Probabilty of False Alarm (PFA) is estimated as the fraction of times the NN output indicates that the signal is FM when the input is actually BWGN. Furthermore, the Probabilty of Error (PE) is estimated as the fraction of times the NN indicates incorrectly what the signal is no matter if it is an FM signal or BWGN. Notice that these probabilities are not directly dependent on a voltage threshold determined from the SNR comparison. Rather they depend on the sigmoid transfer function threshold which is found based on NN training.

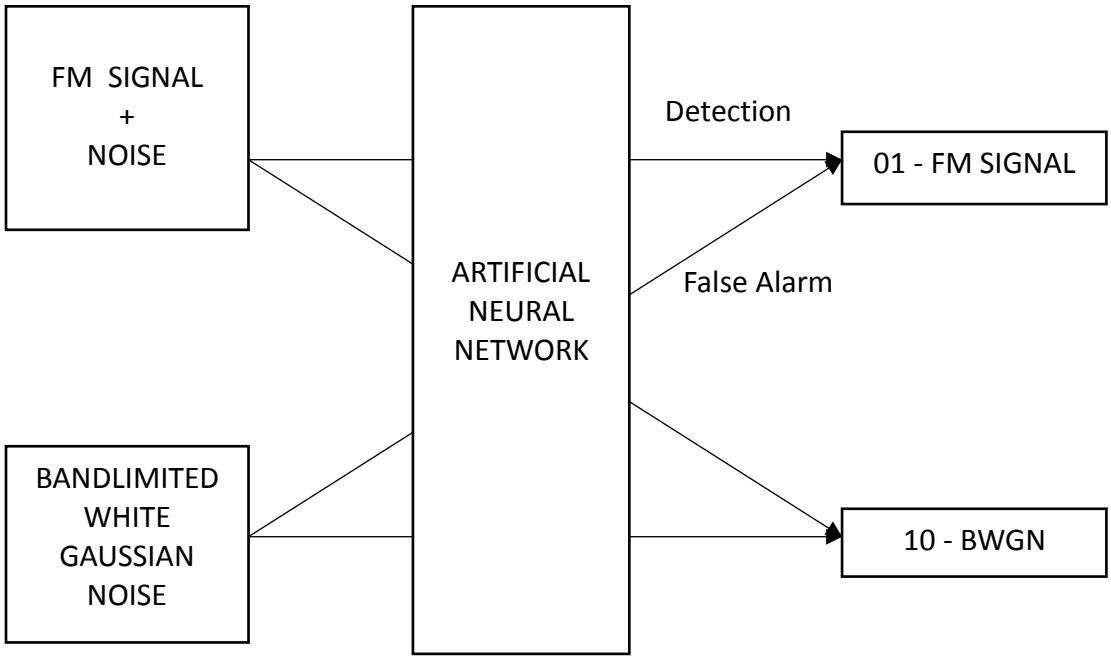


Figure 4.1 Probability of detection and probability of false alarm diagram

Early Testing

Initially, we were inclined to maximize the amount of pre-processing so as to feed a minimum number of signal attributes to the NN. This idea is to compute time and frequency moments of the signal. In the time domain, we define the nth time-moment as

$$E[t^n] = \int_0^T t^n |s(t)|^2 dt \tag{12}$$

where $|s(t)|$ is the normalized envelope of the signal and T is the signal duration. The integral is performed over the duration of the signal. Consequently, mean time and root mean square (rms) time are defined, respectively, as

$$\mu = E[t] \tag{13}$$

and

$$t_{rms} = \sqrt{E[(t-\mu)^2]} \tag{14}$$

Two additional statistics of interest are the skewness

$$\gamma = E\left[\left(\frac{t-\mu}{t_{rms}}\right)^3\right] \tag{15}$$

and the kurtosis

$$k = E\left[\left(\frac{t-\mu}{t_{rms}}\right)^4\right] \tag{16}$$

In the frequency domain, the nth moment is given as

$$E[f^n] = \int_0^\infty f^n |S(f)|^2 df \tag{17}$$

where $|S(f)|^2$ is the power spectrum of the signal. The mean frequency is now defined as

$$\mu_f = E[f] \tag{18}$$

and the root mean square bandwidthy is given by

$$\beta_{rms} = \sqrt{E\left[\left(f - \mu_f\right)^2\right]} \tag{19}$$

To calculte the skewness of the time moment we use

$$\gamma_f = E\left[\left(\frac{f-\mu_f}{\beta_{rms}}\right)^3\right] \tag{20}$$

Finally to obtain the kurtosis of the time moment we use

$$Kurtosis = E\left[\left(\frac{f-\mu_f}{\beta_{rms}}\right)^4\right] \tag{21}$$

Experimentally, we assumed that the envelope of the complex signal was rectangular. Consequently, regardless of the type of jamming signal, the time moments were essentially invariant as shown in Table 4.1. It should be noted that these statistics were averaged over twelve experiments to reduce their measurement variance.

Table 4.1. Time and frequency domain moments of jamming signals.

Time Moments			
	LFM	**PFM**	**BWGN**
Mean	8.2	8.2	8.7
RMS	4.7	4.7	4.7
Skewness	0.0	0.0	0.0
Kurtosis	1.8	1.8	1.8

Frequency Moments			
	LFM	**PFM**	**BWGN**
Mean	125.0	112.5	31.5
RMS	14.4	14.2	97.4
Skewness	0.3	1.1	3.2
Kurtosis	2.5	4.9	12.6

In a separate experiment, the input matrix was composed of signal sequences in the time-domain. Figure 4.2 illustrates the results for M=1024 signals, each having N=1024 samples, where the probability of detection (PD) and the probability of false Alarm (PFA) were calculated directly from the confusion matrix provided by the NN. The PFA (purple dots) and the PD (blue dots) are shown as a function of SNR. The results clearly indicate that the NN is unable to differentiate between a LFM signal and BWGN.

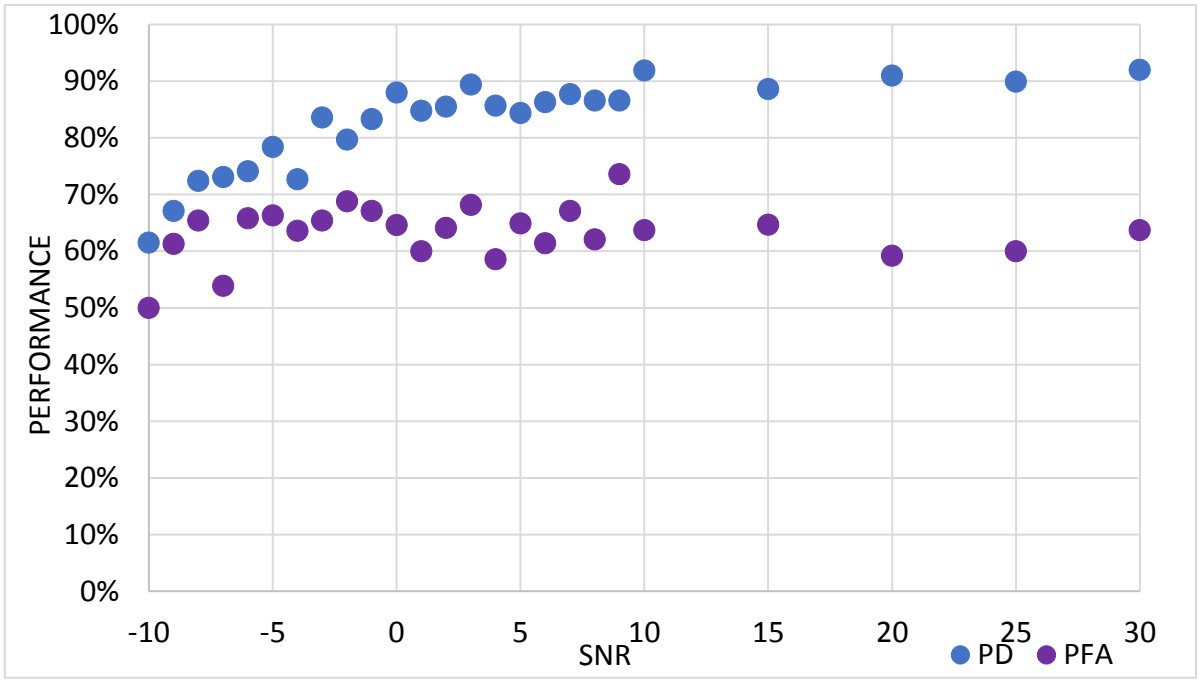


Figure 4.2. Neural network performance when the input data consisted of LFM and BWGN sequences (N=1024, M=1024). Although the PD might be acceptable, the PFA is extremely high.

We conducted further tests by increasing the number of signals and samples per signal hoping to see an improvement. The number of signals and samples per signal are crucial for training the NN. Insufficient data lead to poor outcomes; having too many data points can cause overfitting. An experiment was carried out to assess the performance of

the NN for the binary test LFM vs. BWGN at an SNR of -5 dB. Table 4.2 shows the probability of error for multiple values of M and N. Notice that the number of floating point operations is of the order of $M \times N \log_2 N$. Based on these results, we decided to use M= 2048 signals and N=1024 samples per signal regularly. In the table the percent error %E is the overall error of the NN. This error is the fraction of the times the NN classified the LFM and BWGN jamming signals wrong.

Table 4.2. Data generated to test the desired number of samples and signals used.

SNR=-5 dB							
N=256		N=512		N=1024		N=2048	
M	%E	M	%E	M	%E	M	%E
256	40.43%	256	50.00%	256	41.21%	256	32.62%
512	36.04%	512	47.56%	512	50.68%	512	47.56%
1,024	45.61%	1,024	42.58%	1,024	28.76%	1,024	29.30%
2,048	40.20%	2,048	36.72%	2,048	29.54%	2,048	23.00%
4,096	36.61%	4,096	34.89%	4,096	29.21%	4,096	24.83%
8,192	35.46%	8,192	33.25%	8,192	30.68%	8,192	26.43%

Final Design

Based on early designs, we decided to require some level of (signal) preprocessing to construct an input matrix that could be used effectively by the NN to classify the jamming signals. Furthermore, we chose to compare two approaches, one based on the power spectrum and another based on the autocorrelation of the jamming signals.

Figure 4.3 shows the general approach used to populate the training matrix with

the power spectrum of signals. In this case, the spectra of bandlimited filtered signals are estimated using the DFT. A total of M spectra is computed and stored in the training matrix. For the binary testing case, half of the spectra are bandlimited noise and the other half are the spectra of FM signals. The NN provides a confusion matrix from which the PD and PFA are estimated.

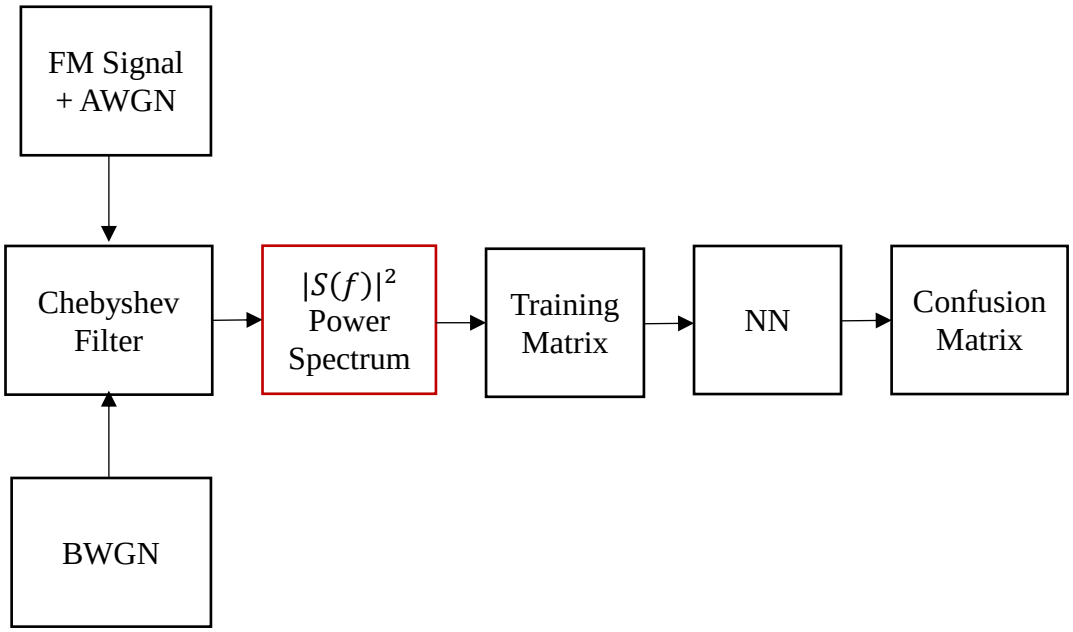


Figure 4.3. Block diagram of neural network classifier in which the training matrix is a power spectra set.

Figure 4.4 shows the training performance for the power spectrum when the inputs are the LFM and BWGN jamming signals. The training set of spectra was arranged as an MxN (2048, 1024) matrix. The performance is shown against SNR. In this case, the PD monotonically increases as a function of SNR and is nearly optimal (100%) for SNR greater than 5 dB. Similarly, the PFA is near 0% for SNR >0 dB. These probabilities deteriorate as the SNR decreases. Below –7 dB, the NN is unable to correctly identify

the LFM signal better than a random flip of a coin.

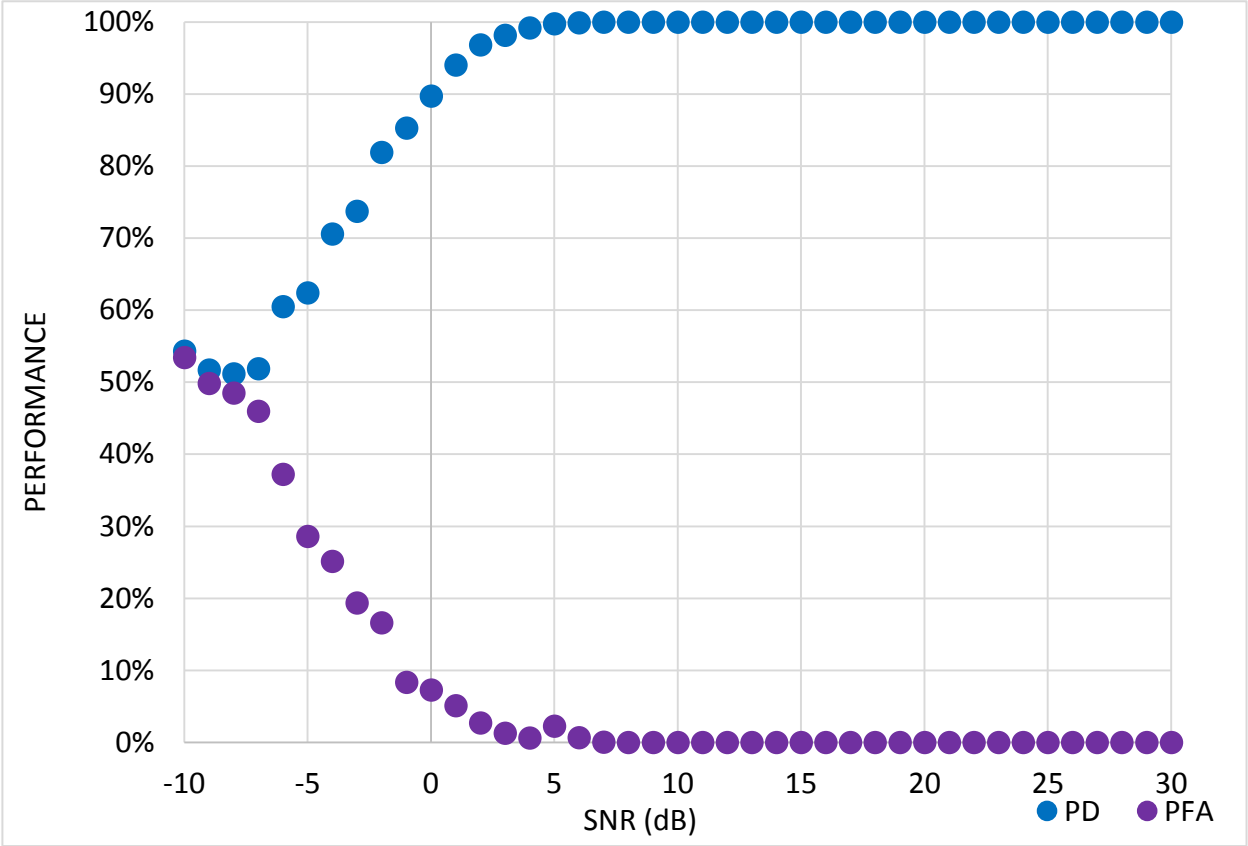


Figure 4.4. Training performance using power spectrum for linear FM and BWGN.

Figure 4.5 shows the general approach used to populate the training matrix with the autocorrelation of signals. In this case, the autocorrelation is computed after the signal is filtered. A total of M autocorrelations are computed and stored in the training matrix. For the binary testing case, half of the autocorrelations are for bandlimited noise and the other half are for FM signals. The NN provides a confusion matrix from which the

PD and PFA are estimated.

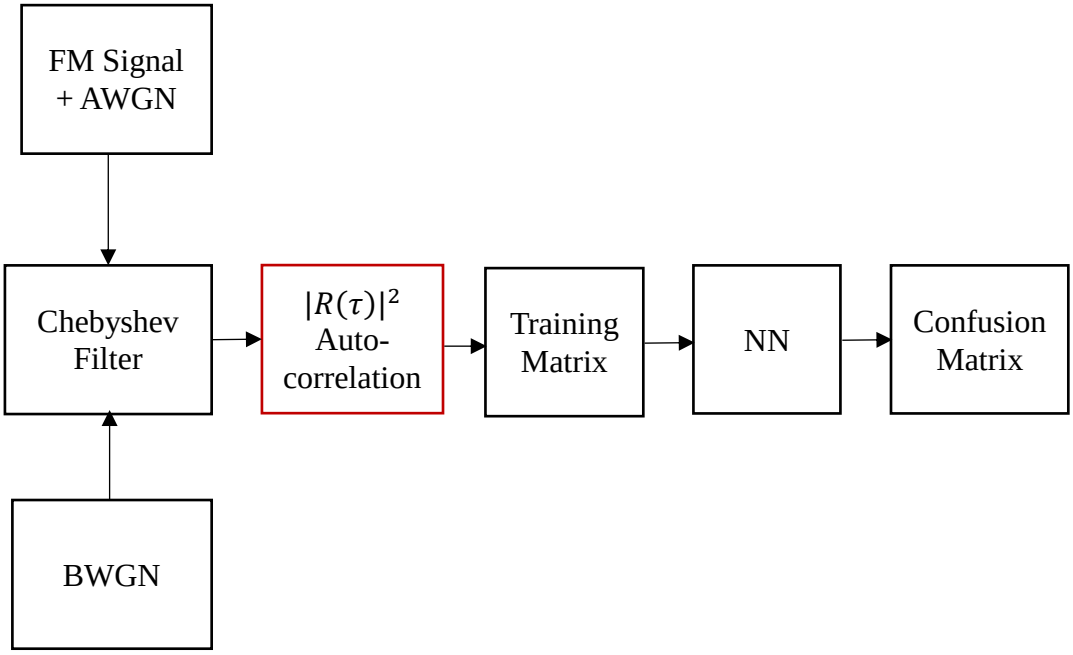


Figure 4.5. Block diagram of neural network classifier in which the training matrix is a set of autocorrelations.

Figure 4.6 shows the training performance for the autocorrelation for the binary hypothesis test LFM vs. BWGN. The autocorrelations used for training were arranged as an MxN (2048, 1024) matrix. Performance is shown as a function of SNR. The NN performs nearly perfectly providing a PD of 100% and a PFA of 0% for SNR greater than 3dB. These probabilities deteriorate as the SNR decreases and a practical threshold could be set at 0 dB. However, it is worth noticing that for SNR as low as -10 dB, the PD

is slightly better than 50%.

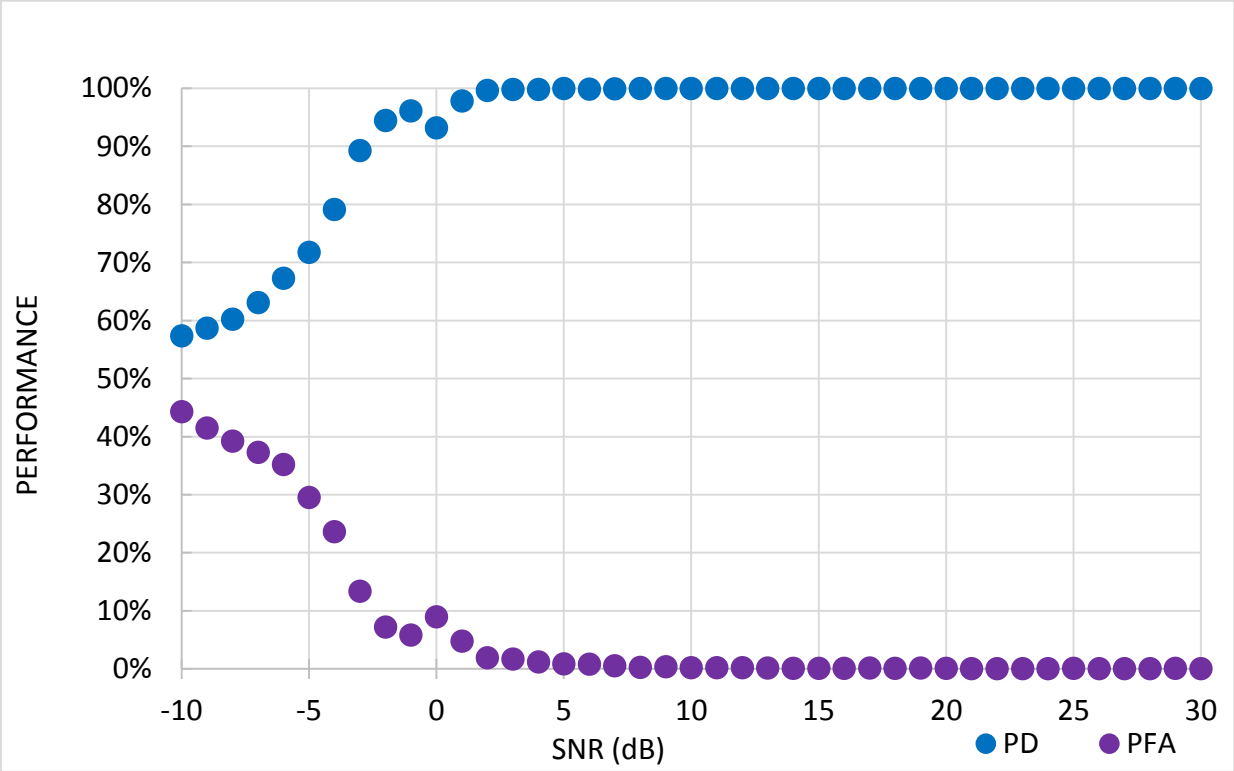


Figure 4.6. Training performance using autocorrelation for LFM and BWGN classification

Another test was conducted to determine if the NN could differentiate between two FM signals with substantially different instantaneous frequencies. Figure 4.7 and 4.8 illustrate the results obtained for this test. As it can be observed, regardless of whether the inputs were spectra or autocorrelations, the NN was able to correctly classify the LFM and PFM signals at the same rate. Furthermore, the NN is able to tell two different types of FM signals apart better than LFM and BGWN.

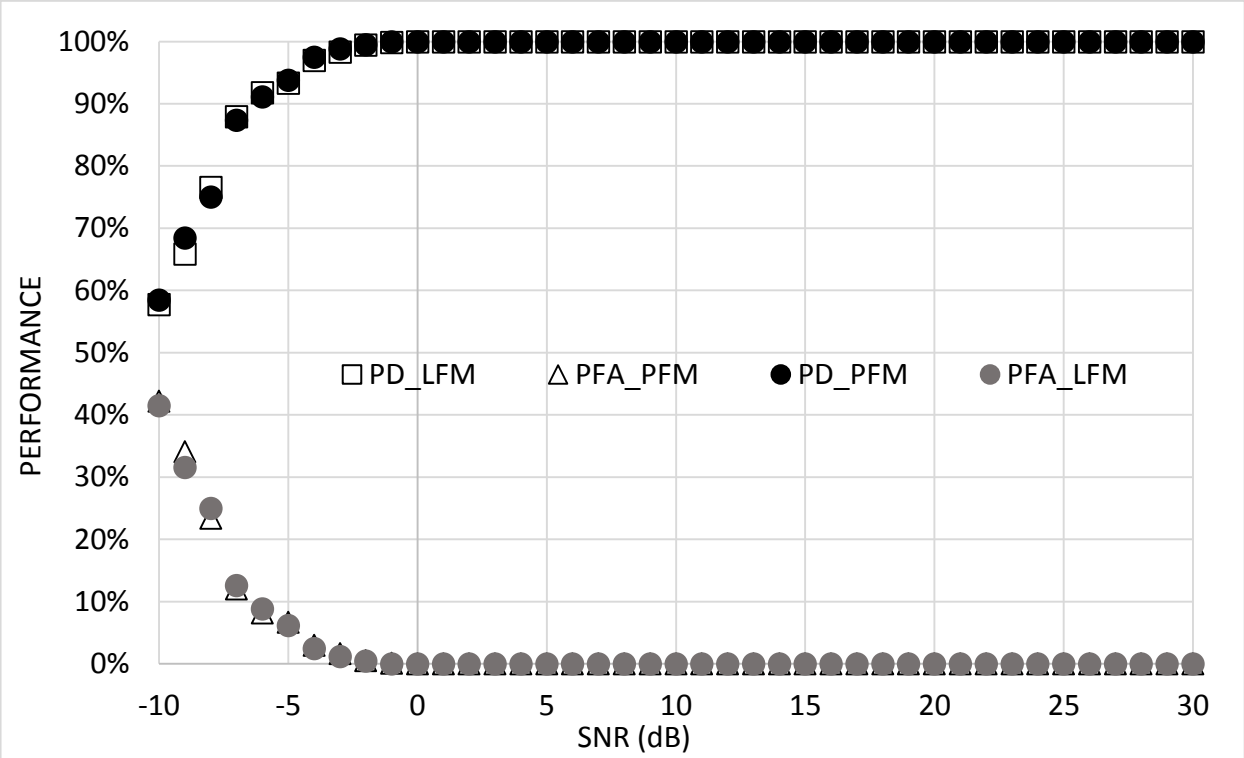


Figure 4.7. Training performance using power spectrum for LFM and PFM classification

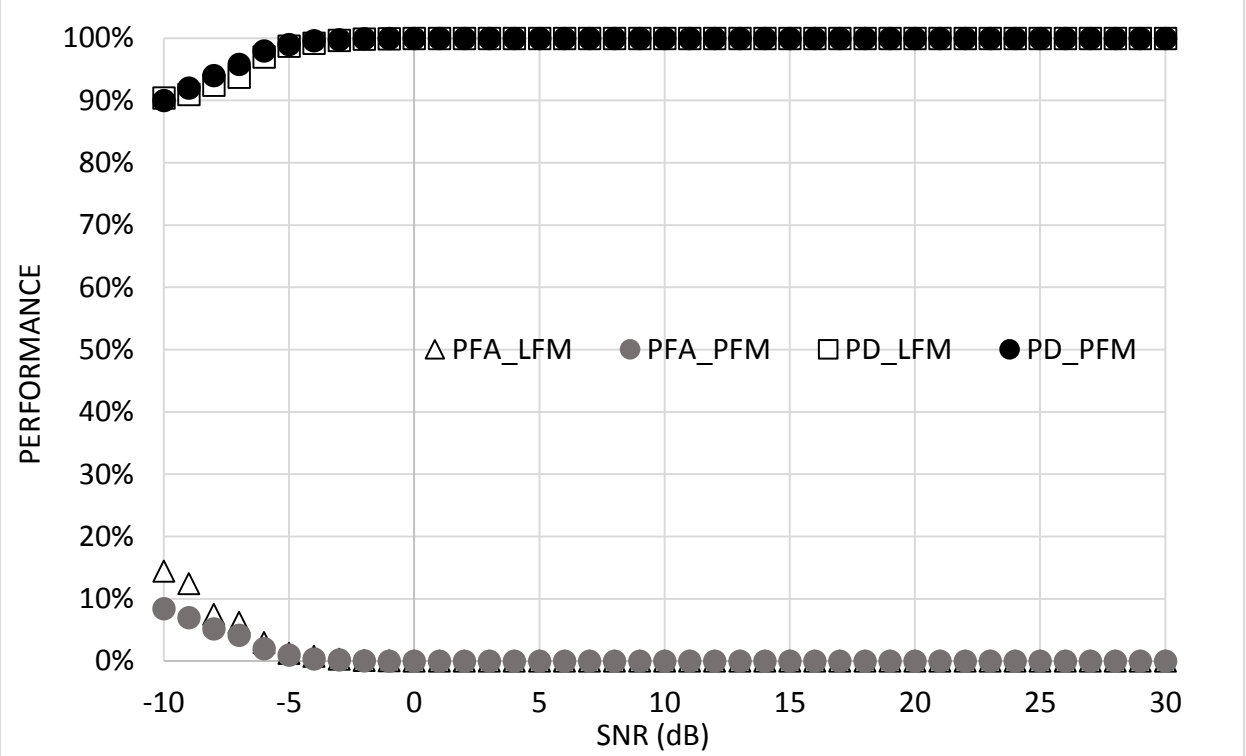


Figure 4.8. Training performance using autocorrelation for LFM and PFM classification

When trying to identify PFM and BGWN, we encounter better results than in the previous case. For this experiment, we used M = 2048 and N= 1024. In Figure 4.9, the NN achieves an 88% PD and a 10% PFA at -5 dB SNR when using power spectra as input. In contrast, in Figure 4.10, the NN reaches a 98% PD and a 2.30% PFA at -5 dB SNR. Using autocorrelation preprocessing yields significantly better results than using power spectrum preprocessing.

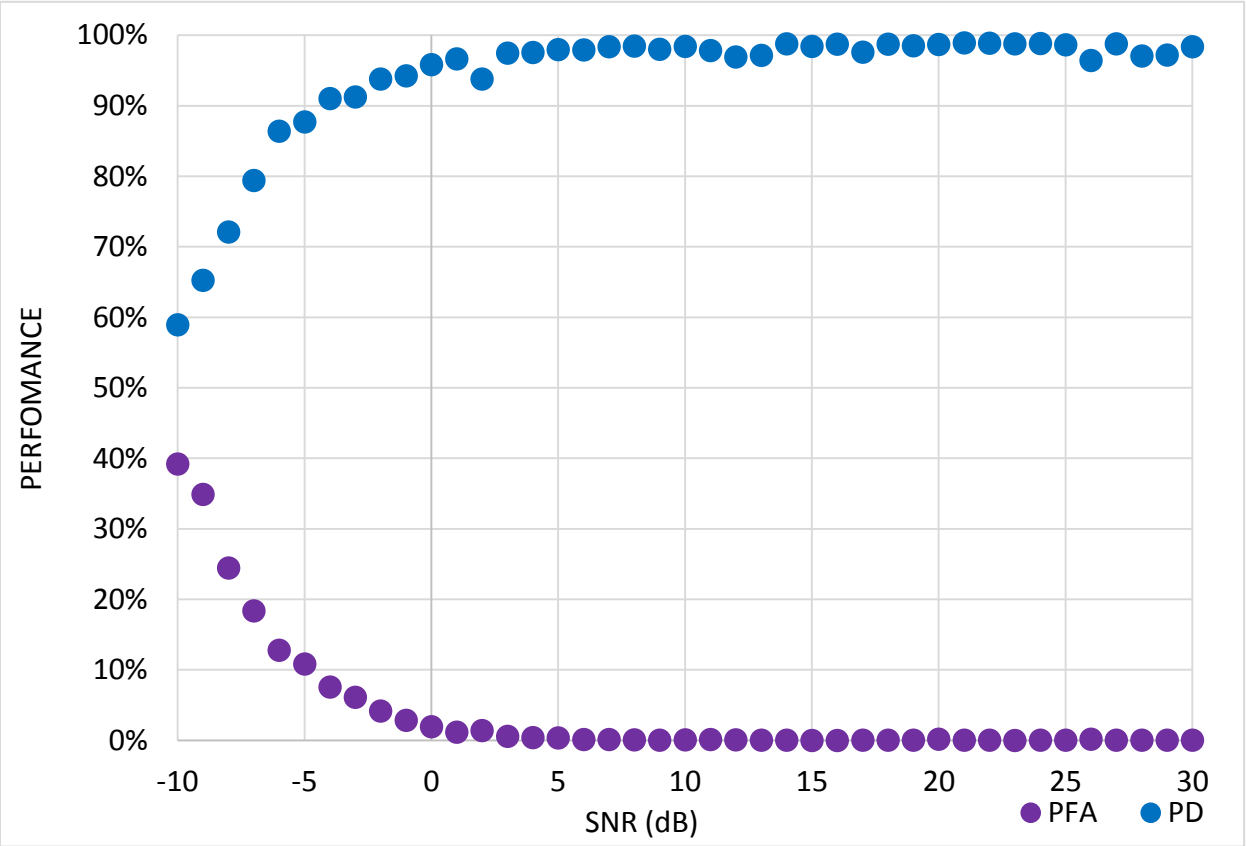


Figure 4.9. Training performance using the power spectrum to identify a power-Law FM from a jammer signal.

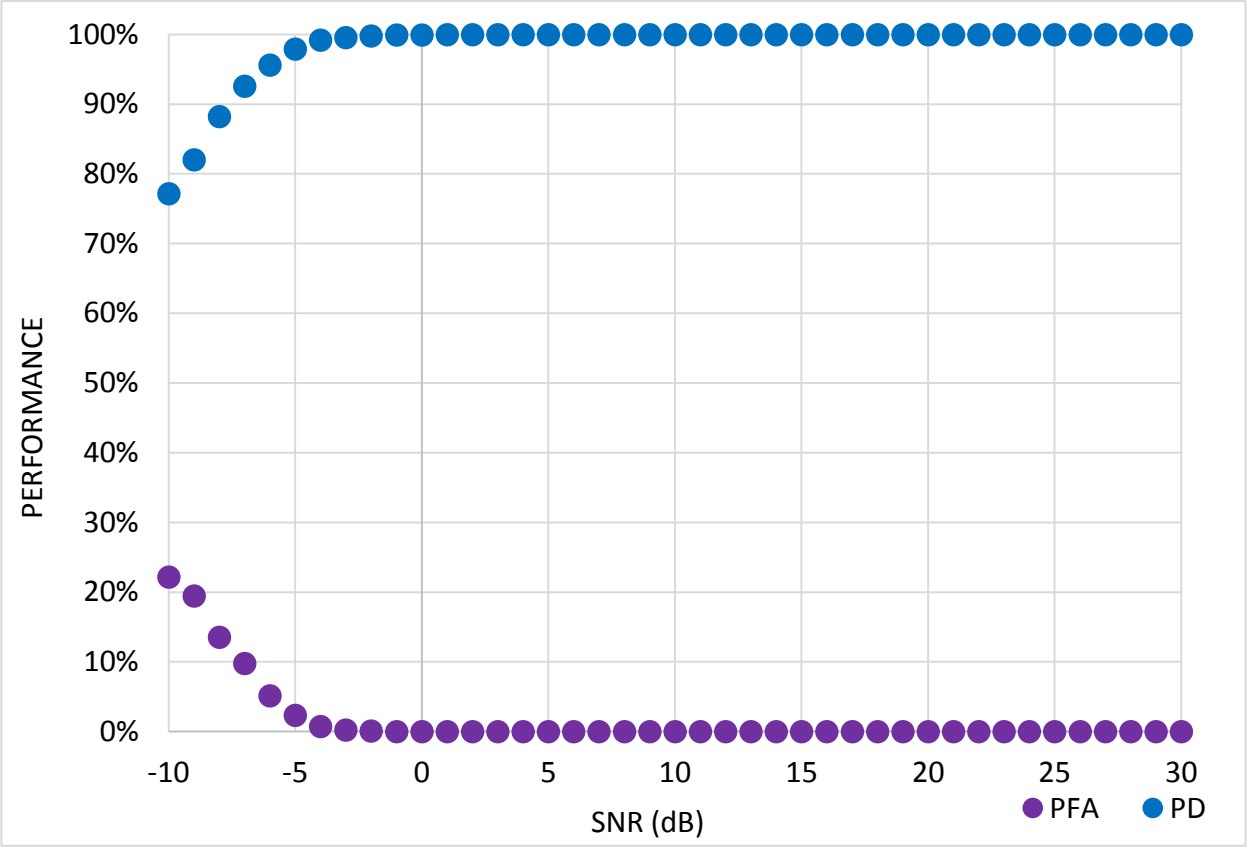


Figure 4.10. Training performance using the autocorrelation to classify a power-law FM from a jammer signal.

As an extreme case, we consider and SNR of -5dB to draw some conclusions for this threshold. When training the NN with LFM and BWGN using the power spectra as input, we obtained a PD of 62% and a PFA of 29%. Using the autocorrelation method instead, we obtain a PD of 72% and a PFA of 29%. Clearly the autocorrelation method yields better results.

Finally, we consider the case of triple hypothesis testing. Here, the NN is meant to differentiate from among three different types of bandlimited signals. Performance is measured using the false positive percentage. This test takes the average of twelve experiments per point to reduce measurement variances, with M=2048 and N=1024. In

Figures 4.11 and 4.12 it can be observed that the autocorrelation has a 20% lower starting false positive than the spectrum. Also, the PFM shows convergence to 0% false positive faster than the LFM.

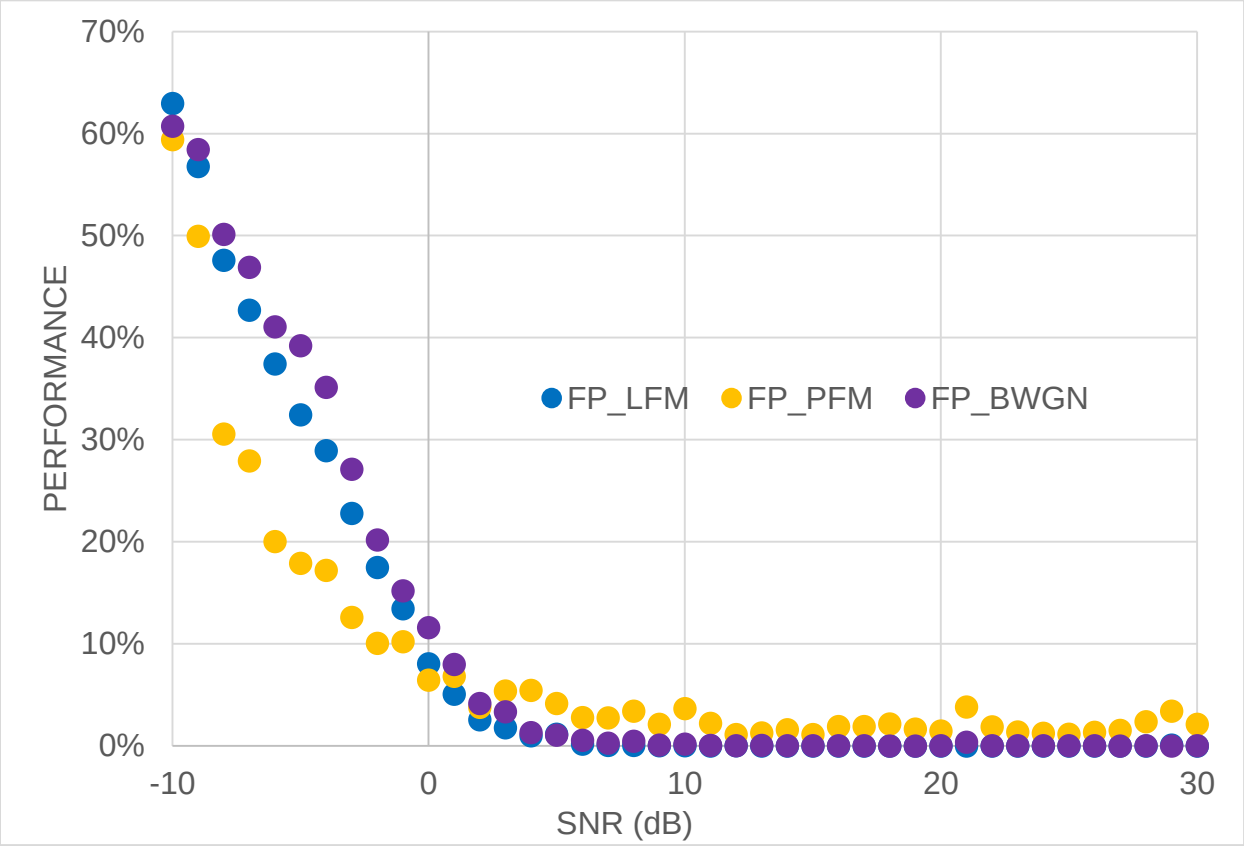
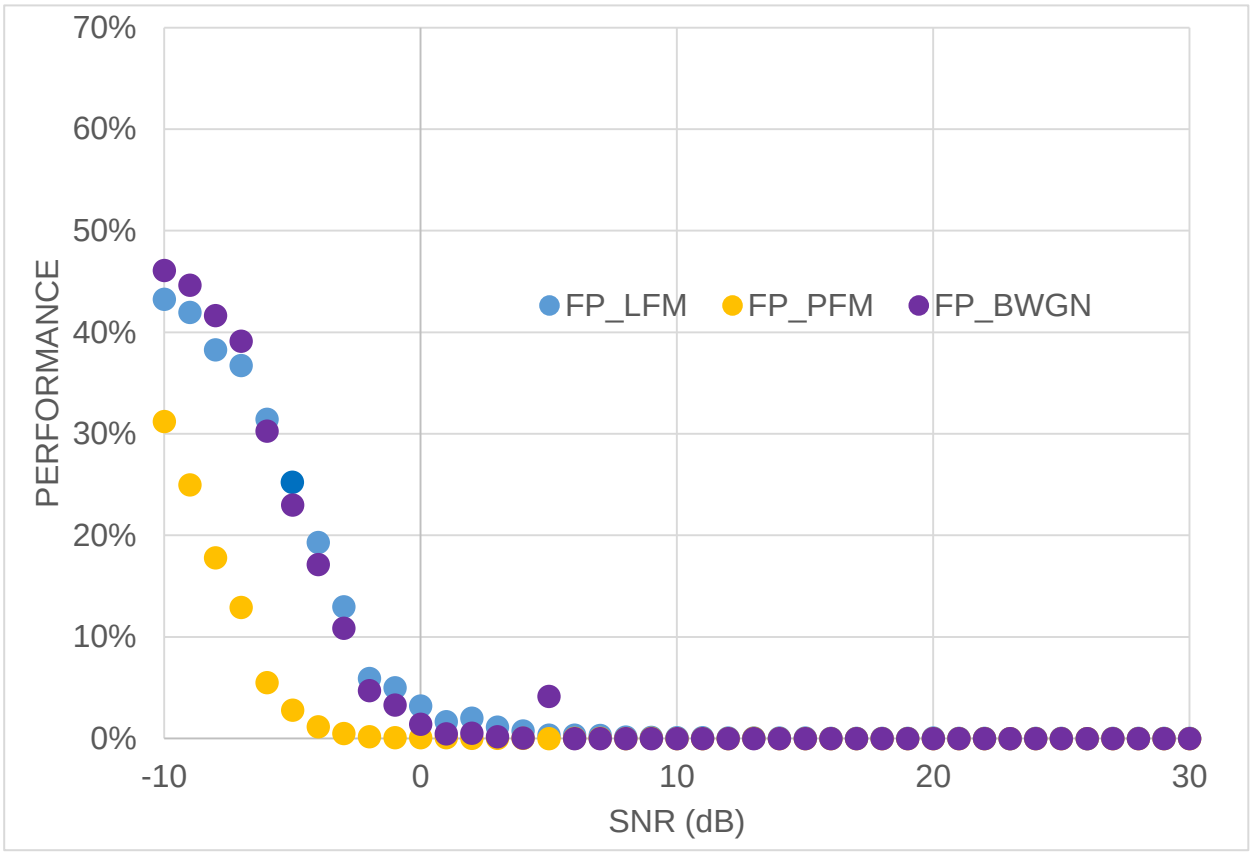


Figure 4.11 Training performance using the power spectrum to estimate false positives



.

Figure 4.12 Training performance using the autocorrelation to estimate false positives.

Chapter 5: Conclusion

This thesis presented a systematic approach with which to train and test an artificial neural network to classify three types of jamming signals: LFM, PFM, and BWGN. A preliminary study was conducted to determine if the NN would be able to identify these signals using time moments or frequency moments. For instance, when time-domain sequences were fed to the NN, the resulting PD and PFA measurements were unacceptable. The PD was close to 100% but at the same time the PFA was above 60%. The problem was solved by introducing a preprocessing stage that fed normalized power spectra to the NN. High PD and low PFA were achieved by increasing the number of spectra as well as the number of samples. However, the idea of using data in the time lingered and it was decided to use the normalized autocorrelation of signals as the input to the NN. Interestingly, the NN performed better in all the cases when the autocorrelation was used instead of the power spectrum. For instance, when differentiating LFM from BWGN the autocorrelation method was better at identifying the signals. In another experiment, the NN was challenged to classify PFM and BWGN. Results from this test indicated that the PFM is more identifiable than the LFM when compared to BWGN. In yet another experiment, the NN was used to differentiate between two types of FM signals. In this trial, both the spectrum and the autocorrelation methods were able classify the signals correctly with the same accuracy. For the last experiment, the NN was fed autocorrelation or spectra data from LFM, PFM, and BWGN signals. Once again, the autocorrelation demonstrated superior performance.

From a theoretical viewpoint, the power spectrum and the autocorrelation of a signal carry the same amount of information so there should be no obvious reason why

a NN suited with autocorrelation preprocessing should perform better than a NN with spectrum preprocessing. This is clearly true under high SNR conditions. However, when the SNR is reduced, there appears to be an advantage in compressing the signal in the time domain. Perhaps the structure of the autocorrelation mainlobe and sidelobes is preserved sufficiently at low SNR to allow the NN to classify signals correctly. This observation is, of course, speculative in nature and biased by our human way of visualizing and interpreting data. All we can say for now is that the results stand on their own. Further research could focus on some attributes of the autocorrelation and spectra to determine if additional preprocessing yields even better results.

References

1. Kahn, D. (1991). The Intelligence Failure of Pearl Harbor. *Foreign Affairs, 70*(5), 138-152. doi:10.2307/20045008
2. Carter, Christopher A.P., and Nathalie Mass. *Neural Networks for Classification of Radar Signals*. 1st ed. Ottawa: N.p., 1993. Web. 7 Oct. 2016.
3. Ahalt, S. C. Et al. *The performance of Synthetic Neural Network Classification of Noisy Radar Signals.* 1st ed. Columbus, Ohio: The Ohio State University, 1989. Web. 7 Oct. 2016.
4. Cheikh, K. And F. Soltani. *Application Of Neural Networks To Radar Signal Detection In K-Distributed Clutter*. IEE Proceedings - Radar, Sonar and Navigation 153.5 (2006): 460-466. Web. 7 Oct. 2016.
5. Chralampidis, D., T. Kasparis, and M. Georgiopoulos. *Classification Of Noisy Signals Using Fuzzy ARTMAP Neural Networks*. IEEE Trans. Neural Networks. 12.5 (2001): 1023-1036. Web.
6. Darzikolaei, Mohammad Akhondi, Ata Allah Ebrahimzade, and Elahe Gholami. *Classification Of Radar Clutters With Artificial Neural Network.* 2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI) (2015): 577-581. Web. 7 Oct. 2016.
7. Ibrahim, N.K., Raja Abdullah, and M.I. Saipan. *Artificial Neural Network Approach in Radar Target Classification.* 1st ed. Serdang Selangor, Malaysia: University Putra Malaysia, 2009. Web. 7 Oct. 2016.
8. Jordanov, Ivan, Nedyalko Petrov, and Alessio Petrozziello. "Supervised Radar Signal Classification." *2016 International Joint Conference on Neural Networks (IJCNN)* (2016): n. page. Web. 6 Jan. 2017.
9. Jarabo-Amores, P. Et al. A *Neural Network Approach To Improve Radar Detector Robustness.* 1st ed. Madrid: Escuela Politecnica Superior, Universidad de Alcala, 2006. Web. 7 Oct. 2016.
10. Darzikolaei, Mohammad Akhondi, Ata Allah Ebrahimzade, and Elahe Gholami. *Classification Of Radar Clutters With Artificial Neural Network.* 2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI) (2015): 577-581. Web. 7 Oct. 2016.
11. I. Kanellopoulos and G. Wilkinson, *Strategies and best practice for neural network image classification*, International Journal of Remote Sensing, vol. 18, no. 4, pp. 711-725, 1997.
12. Jain A., Jianchang Mao and K. Mohiuddin, "Artificial neural networks: a tutorial," *Computer*, vol. 29, no. 3, pp. 31-44, 1996.
13. J. Dayhoff and J. M. DeLeo., "Artificial Neural Networks Opening the Black Box," *Conference on Prognostic Factors and Staging in Cancer Management: Contributions of Artificial Neural Networks and Other Statistical Methods*, pp. 1-21, 2001.
14. P. Sydenham and R. Thorn, *Handbook of measuring system design*, 1st ed. Chichester, England: Wiley, 2005, pp. 901-910.
15. A. Soto, A. Mendoza and B. Flores, "Optimization of Neural Network Architecture For Classification of Radar Jamming FM Signals," *SPIE*, 2017.

16. M. Beale, M. Hagan and H. Demuth, *MATLAB The Neural Network Toolbox*. Natick, MA 01760-2098: The MathWorks, Inc., 2017. (version 4)
17. M. Hudson Beale, M. Hagan, and H. B. Demuth, "Neural network performance - MATLAB cross entropy - the MathWorks United Kingdom," *Mathworks.com*, 2017. [Online]. https://www.mathworks.com/help/nnet/ref/crossentropy.html. [Accessed: 07- Mar- 2017].
18. M. Møller, "A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning," *DAIMI Report Series*, vol. 6, no. 339, pp. 1-9, 1991.

Vita

Ariadna Mendoza was born in Mexico on September 17, 1992. She moved to the United States in 2004. After finishing senior year at Socorro high school, Ariadna studied electrical and computer engineering at The University of Texas at El Paso where she received her Bachelors of Science in 2015. The following semester she enrolled again at The University of Texas at El Paso where she is expected to graduate in May 2017 with a Master's of Science in Electrical Engineering.

Contact Information:aemendoza@miners.utep.edu

This thesis was typed by Ariadna Mendoza