DEEP LEARNING IN ELECTRONIC WARFARE SYSTEMS: AUTOMATIC PULSE DETECTION AND INTRA-PULSE MODULATION RECOGNITION

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF SCIENCE

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

By

Fatih Cagatay Akyon

December 2020

Deep Learning in Electronic Warfare Systems: Automatic Pulse Detection and Intra-pulse Modulation Recognition
By Fatih Cagatay Akyon
December 2020

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Orhan Arikan(Advisor)

Sinan Gezici

Abdullah Aydin Alatan

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

DEEP LEARNING IN ELECTRONIC WARFARE SYSTEMS: AUTOMATIC PULSE DETECTION AND INTRA-PULSE MODULATION RECOGNITION

Fatih Cagatay Akyon
M.S. in Electrical and Electronics Engineering
Advisor: Orhan Arikan
December 2020

Detection and classification of radar systems based on modulation analysis on pulses they transmit is an important application in electronic warfare systems. Many of the present works focus on classifying modulations assuming signal detection is done beforehand without providing any detection method. In this work, we propose two novel deep-learning based techniques for automatic pulse detection and intra-pulse modulation recognition of radar signals. As the first nechnique, an LSTM based multi-task learning model is proposed for end-to-end pulse detection and modulation classification. As the second technique, re-assigned spectrogram of measured radar signal and detected outliers of its instantaneous phases filtered by a special function are used for training multiple convolutional neural networks. Automatically extracted features from the networks are fused to distinguish frequency and phase modulated signals. Another major issue on this area is the training and evaluation of supervised neural network based models. To overcome this issue we have developed an Intentional Modulation on Pulse (IMOP) measurement simulator which can generate over 15 main phase and frequency modulations with realistic pulses and noises. Simulation results show that the proposed FFCNN and MODNET techniques outperform the current state-of-the-art alternatives and is easily scalable among broad range of modulation types.

Keywords: intra pulse modulation, electronic warfare, convolutional neural network (CNN), long short term memory (LSTM), deep learning, machine learning, multi task learning, simulator, feature fusion, time frequency analysis, robust least squares, pulse detection, modulation classification, waveform recognition, sincnet, energy detector, autoencoder.

ÖZET

ELEKTRONİK TAARRUZ SİSTEMLERİNDE DERİN ÖĞRENME: OTOMATİK DARBE TESPİTİ VE İSTEMLİ DARBE İÇİ KİPLEME SINIFLANDIRMA

Fatih Cagatay Akyon
Elektrik-Elektronik Mühendisliği, Yüksek Lisans
Tez Danışmanı: Orhan Arıkan
Aralık 2020

Radarın ilettikleri darbelere göre tespiti ve sınıflandırılması elektronik harp sistemlerinde önemli bir uygulamadır. Mevcut çalışmaların çoğu, herhangi bir darbe tespiti yöntemi sunmadanö önceden sinyal darbesi tespitinin yapıldığını varsayarak modülasyonların sınıflandırılmasına odaklanır. Bu çalışmada, radar sinyallerinin otomatik darbe tespiti ve istemli darbe içi kipleme sınıflandırma için iki yeni derin öğrenme tabanlı teknik öneriyoruz. İlk yaklaşımda, ölçülen radar sinyalinin yeniden tayin edilmiş spektrogramı ve özel bir fonksiyon tarafından filtrelenen anlık fazlarının saptanan aykırı değerleri, çoklu evrişimli sinir ağlarını eğitmek için kullanılır. Ağlardan otomatik olarak çıkarılan özellikler, frekans ve faz modülasyonlu sinyalleri ayırt etmek için birleştirilir. İkincisinde, uçtan uca darbe tespiti ve modülasyon sınıflandırması için UKSB tabanlı çok görevli bir öğrenme modeli önerilmiştir. Bu alandaki başka önemli sorun, denetimli sinir ağı tabanlı modellerin eğitimi ve değerlendirilmesinde kullanılacak kamuya açık radar darbe verilerinin eksikliğidir. Bu sorunun üstesinden gelmek için gerçekçi darbeler ve gürültüler içeren 15'ten fazla ana faz ve frekans modülasyonu oluşturabilen bir IMOP ölçüm simülatörü geliştirdik. Simülasyon sonuçları, önerilen FFCNN ve MODNET tekniklerinin mevcut son teknoloji alternatiflerden daha iyi performans gösterdiğini ve çok çeşitli modülasyon türleri arasında kolayca ölçeklenebilir olduğunu göstermektedir.

Anahtar sözcükler: darbe içi kipleme, elektronik taarruz, evrişimsel sinir aği, uzun kısa süreli bellek, derin öğrenme, makine öğrenmesi, çok görevli öğrenme, simülasyon, öznitelik füzyonu, zaman frekans analizi, gürbüz en küçük kareler, darbe tespiti, kipleme sınıflandırma, dalga biçimi tanıma, sincnet, enerji tespitçisi, oto kodlayıcı.

Acknowledgement

First of all, I would like to express my deepest gratitude to my supervisor Prof. Dr. Orhan Arikan for his continuous support, guidance, patience, encouragement in the path of creation of this thesis. I learnt a lot not only in the perspective of research but also the ethical concerns and the administration. I thank him for his contribution to my scientific vision and my writing skills. I am very grateful for his immediate responses when I need some help in anything I have asked for and the great positive communication and understanding which he provides for me.

Secondly, I would like to present my special thanks to my supervisors Yasar Kemal Alp, Gokhan Gok and Fatih Altiparmak who guided me in the field on electronic warfare during my time at Aselsan, Electronic Warfare and Intelligence Systems Division.

I also would like to present my special thanks to my family who always supported me in my decisions. The biggest portion of the thank belongs to my beloved mom. She always take care of me whenever and wherever I need help without considering herself. Secondly, I want to present my thanks to my father who has significant guidance in my decisions and always gives the freedom in my choices. Finally, thanks to my twin siblings for being my best friends when I needed them.

My wife Seyma Handan deserves my most special thanks for being the motivation behind the thesis with her endless love and support.

I also want to present my special thanks to my friends Esat Kalfaoglu, Omer Cem Akyol, Cemil Cengiz and Volkan Dinc for their sincere friendship and their academic guidance on the thesis.

This work is supported by TÜBİTAK within the scope of 2210/A scholarship.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Automatic pulse support detection and classification of amplitude, phase and frequency modulations present in complex time series signals plays a pivoted role in communication and signal processing fields. It has a great significance for communication applications such as, operator regulation, communication anti-jamming, and user identification [1]. It is also useful in Electronic-Warfare systems while automatically classifying radar intra-pulse modulation types [2], [3].

Before the emergence of more sophisticated approaches, most of the modulation classification methods were based on two major phases: feature extraction and classification. In [4] and [5], automatic classification of communication signals using an SVM based classifier are proposed. [4] transforms the input signals to a high dimensional feature space using wavelet kernel function before feeding them into an SVM classifier. On the other hand, four Wavelet based noise insensitive features are extracted in [5] before getting the classification results using an SVM-DDAG (Decision Directed Acyclic Graph based Support Vector Machine). The method proposed in [4] is evaluated on two modulation types (analogue modulated signals and digitally modulated signals) while [5] is tested on eight different modulation types achieving an overall success rate of 95% at a signal-to-noise ratio (SNR) of 10dB.

[6], [7], [8], and [9] also focus on intra-pulse modulation classification with traditional feature extraction based methods. In [6], an automatic intra-pulse recognition system is introduced that is aimed to be used in various spectrum management, surveillance and cognitive radio or radar applications. The proposed method is evaluated on eight classes of frequency and phase modulations. Extraction of features based on Wigner and Choi–Williams time-frequency distributions are proposed. Then a multi layer perceptron structure is employed as the classifier. Simulation results show that the classification system achieves an overall correct classification rate of 98% at an SNR of 6 dB on data similar to the training data. In [7], 23 features based on time-frequency image, second order statistics, power spectral density, and instantaneous properties of the input signal are extracted to classify low probability of intercept (LPI) radar modulations using a recurrent network named Elman neural network. Eight modulation types are used to test the proposed technique with an overall ratio of successful recognition of 94.7% at an SNR of –2 dB. On the other hand, [8] proposes a new intra-pulse modulation classification algorithm based on auto-correlation function (ACF) and directed graphical model (DGM). The ACFs are calculated from analytic radar signals are to make the discrimination of modulation types more obvious. Four features are extracted from the denoised ACF and A DGM is used to represent the joint probability distribution of the four features along with the category and to classify unknown modulation types. Simulation results on three modulations show 90% accuracy at an SNR of –10 dB is achieved with the proposed method. Unlike previous methods, [9] focuses on the problem of effective classifier selection. Authors present a boosting algorithm as an ensemble frame to achieve a higher accuracy than a single classifier for the modulations of communication signals. Five kinds of entropy are extracted from the signals as the features. Then, AdaBoost algorithm based on decision tree is utilized to confirm the idea of boosting algorithm. To evaluate the effect of boosting algorithm, eight common communication modulation types are tested at different SNR levels ranging from -10 to 20 dB. Performance of three diverse boosting members is compared by experiments. Reported results indicate that the gradient boosting has better behavior than AdaBoost, and xgboost creates the best results.

The aforementioned classical two-step recognition methods suffer a common weakness. In order to obtain the discriminative features, too much attention have to be paid to discover efficient distinctions and to find ways to characterize them. However, sometimes some important features may be undiscovered, or the extracted features are not discriminative enough for recognition. In particular it has been shown that relatively simple convolutional neural networks (CNN) outperform algorithms with decades of expert feature searches for radio modulation [10]. In this work, authors compare the effect of radio modulation classification using naively learned features against using expert feature based methods such as CNNs. They show significant performance improvements over the traditional methods. [10] shows that blind temporal learning on large and densely encoded time series using deep CNN is viable and a strong candidate approach for this task especially at low signal to noise ratios. Then more related work is presented that focus on deep neural networks to automatically extract the most meaningful features from the raw data and perform modulation classification accordingly.

Sparsifying autoencoder structures is employed in [11] to classify four digital modulation types. As the input of the network, authors compared the effect of three different input structure: 1) in-phase and quadrature constellation points, 2) centroids of the constellation points acquired by applying fuzzy C-means algorithm to in-phase and quadrature constellation points, 3) the high order cumulants up to order 4 of the received samples of each modulation class. The unsupervised learning from these data sets was done using the sparse autoencoders and a supervised softmax classifier was employed for the classification.

Since time-frequency distributions of different modulation types of signals are discriminative, it is reasonable to process raw data in time-frequency domain. Especially in phase and frequency modulated radar signals, the time-frequency images of stable signals contain all the modulation information in terms of radar principles, which can theoretically discriminate the waveform types. For these reasons, some works focused on applying CNNs [12], [2] and convolutional autoencoders (CAE) [13] to the time-frequency images of the modulated signals. [12] uses the time-frequency images (TFI) of the modulated signals that are calculated using the short-time Fourier transform (STFT) as the input of a 13 layer

CNN. Proposed technique is tested on 7 different modulation types having an accuracy of %100 at 0 dB SNR. Similarly [2] uses a CNN to classify time-frequency images of the modulated signals, however it employs Wigner-Ville Distribution (WVD) that has a greater power gain than the STFT while forming the TFIs. Simulation results demonstrate great recognition rate over 7 modulation types under very low SNR conditions. [13] utilizes sparsifying CAE structure to achieve both feature extraction and denoising of the TFIs (using STFT) of the modulated signals with a pretraining. Similar to traditional methods, they develop the feature extractor and classifier as 2 separate structures. Once denoised deep features are extracted from the TFI, they suggest to perform collaborative representation classification to recognize the modulation types. Results show that their suggested method have 90% accuracy over 6 types of modulations at 0 dB SNR. In [14] and [7], Choi-Williams distribution (CWD) is utilized while creating time-frequency images of the raw signals, then a CNN based network is used to further extraction of features and modulation types are given at the output of the network. It has been shown that CWD TFI features are more discriminative than WVD in most of the cases. [15] proposes a novel CNN architecture, called SincNet. In contrast to standard CNNs, that learn all elements of each filter, only low and high cutoff frequencies are directly learned from data with the proposed method. This offers a compact and efficient way to derive a customized filter bank specifically tuned for the desired application. Authors conducted experiments on both speaker identification and speaker verification tasks from raw samples and results were similar to CNN with less number of model parameters.

Common drawback of these works is the lack of a testing dataset consisting of wide range of modulation types, they mostly focus on communication modulations and most of the works that include intra-pulse modulations ignore the classes that are hard to distinguish. Moreover, even tough these works promise modulation classification on low SNR levels all of them assume the pulse/symbol is detected prior to their classification step. Considering that the lowest SNR level required for pulse detection in a typical electronic warfare system is around 10 dB, in order to make modulation classification at a much lower SNR level, a pulse detection technique must first be proposed for that SNR level.

In detection stage, application of matched filter is infeasible since neither the received signal nor its type is known before interception. In addition, general likelihood ratio tests are not also applicable since they require prior knowledge about the modulation type of the received signal [16]. In contrast to aforementioned methods, energy detector is feasable in these systems as it does not have any assumptions about the modulation type, signal shape or arrival time to make a detection. In [17], energy detector is used to detect spread-spectrum signals. It is utilized to detect optimal frequency band sensing time for cognitive radios in [18]. As a drawback, performance of this method gets degraded in low SNR scenarios. Lastly, [19] proposes an autoencoder based signal detection technique.

Our main contribution in this thesis is:

- A new IMOP recognition technique that utilizes time-frequency transformation and convolutional neural network (two conference papers [3, 20], one patent application [21]), and a new end-to-end pulse detection and IMOP recognition technique that utilizes an LSTM based network (one conference paper [22], two patent applications [23, 24]) (Chapter 3)

On top of that more work have been done to analyse the proposed techniques:

- A simulator that can generate more than 15 phase and frequency modulations for realistic noisy radar pulse measurements (Chapter 2).
- Comparison of the proposed models with several baseline detection and recognition techniques (Chapter 4).

Chapter 2

Intentional Modulation on Pulse (IMOP) Simulator

2.1 Motivation

Detection and classification of modulation type of intercepted noisy LPI (Low Probability of Intercept) radar signals in real-time is a vital survival technique required in the electronic warfare systems. Many experts working on radars today are specifying an LPI and low probability of identification (LPID) as an important tactical requirement. The term LPI is that property of a radar that, because of its low power, wide bandwidth, frequency variability, or other design attributes, makes it difficult for it to be detected by means of a passive intercept receiver. In applications such as altimeters, tactical airborne targeting, surveillance, and navigation, the interception of the radar transmission can quickly lead to electronic attack (or jamming) in the case of the parameters of the emitter are determined. Due to the wideband nature of these pulse compression waveforms, however, this is typically a difficult task.

Recently, detection and classification of intra-pulse modulated signals using

neural networks have gained popularity. These techniques first require a supervised training phase which optimizes the model parameters for the distribution present in the training dataset. Once the parameters are found and fixed, inference is performed to predict the unknown modulation type or detect the region where pulse is present. However, the most difficult part is to find a labeled training dataset that has the same distribution as the real-world inference conditions since radar signal measurements with intra pulse modulations are classified and there is not any publicly available dataset on this topic.

To overcome this issue, a simulator that can create realistic examples of phase and frequency modulated pulses have been developed. In the following sections, details on this simulator have been given.

2.2 Signal Model

Noisy, complex baseband samples of a radar pulse $x(t)$ can be modeled:

$$x(t_n) = a(t_n)e^{j\phi(t_n)} + z(t_n) \quad n = 1, 2, ..., N \tag{2.1}$$

where $a(t_n)$ denotes the pulse envelope, $\phi(t_n)$ denotes instantaneous signal phase and $z(t_n)$ denotes zero mean circularly symmetric complex Gaussian noise, t_n denotes the sampling instants and N is the total number of samples. $a(t_n)$ is the instantaneous magnitude of the pulse and it is defined as:

$$a(t_n) = \begin{cases} 1, & T_0 \leq t_n < T_0 + T_g \\ e^{-(t_n - T_0)^2/{\sigma_1}^2}, & t_n < T_0 \\ e^{-(t_n - T_0 - T_g)^2/{\sigma_2}^2}, & T_0 + T_g \leq t_n \end{cases} \tag{2.2}$$

where T_0 is the time of arrival of the pulse, T_g is the pulse width. Note that, the rising and falling regions of $a(t_n)$ are modeled as Gaussian functions with σ_1 and σ_2 variances, respectively.

In Table 2.1, phase and frequency formulations of the modulations that are

implemented in the proposed simulator are given in detail. Here T denotes the pulse duration (assuming the modulation period is also T) in us, $j = 0, 1, 2...k-1$ is the segment number in the stepped frequency waveform, i is the level in a given step; M is the number of frequency steps in polyphase modulations, k is the number of segments in polytime modulations, ΔF is the modulation bandwidth, N_c is the compression ratio in polyphase and polytime code sequence; n is the number of phase states in the code sequence and t is time [25].

In Figures 2.1 and 2.2 instantaneous magnitude, phase and frequency plots can be seen for simulation generated noisy IMOP samples (Frank phase modulated and Costas frequency modulated pulses). 2.1 presents a Frank modulated measurement with 16 μs pulse width and 0.5-1 μs chip duration while 2.2 presents a 5-step frequency hopping pulse of 6.36 μs pulse width and 0.5-1 μs chip duration with 12 MHz frequency deviation. Figure 2.3 presents a QPSK phase modulated IMOP measurement with 100 MHz sampling frequency, 10 μs pulse width and 0.5-1 μs chip duration. Figure 2.4 presents a P1 polyphase modulated IMOP measurement with 100 MHz sampling frequency, 19 μs pulse width, 0.3-0.5 μs chip duration and number of frequency steps of 6 and 7. Figure 2.5 presents a P4 polyphase modulated IMOP measurement with 100 MHz sampling frequency, 13 μs pulse width, 0.3-0.5 μs chip duration and compression ratios of 36 and 49. Figure 2.6 presents a T1 polytime phase modulated IMOP measurement with 100 MHz sampling frequency, 19 μs pulse width, 4 phase states and number of segments as 4, 5 and 6. Figure 2.7 presents a sinusoidal frequency modulated IMOP measurement with 100 MHz sampling frequency, 15 μs pulse width and period and 15 MHz frequency deviation.

2.3 User Interface

In this section screenshots from the user interface of the developed IMOP simulator is provided. GUI is developed using Appdesigner in Matlab2018a. In Figure 2.8 overall screenshot of the IMOP simulator can be seen. Signal parameters can be entered in the top-left part, configured modulation types can be seen on

Table 2.1: Definitions of the Implemented Intra-Pulse Modulations

Modulation type	$\phi_{i,j}[t]\ rad$	$f_{i,j}[t]\ Hz$
SCM	n/a	n/a
LFM	n/a	$f_0 + \frac{\Delta F}{T}t$
Costas FM	n/a	f_j
Sinusoidal FM	n/a	$cos(2\pi\frac{t}{T})$
Triangular FM	n/a	$f_0 + \frac{\Delta F}{T}t$ or $f_0 + \Delta - \frac{\Delta F}{T}t$
K-PSK	$\frac{2\pi j}{K}$	n/a
Frank Code	$\frac{2\pi}{M}(i-1)(j-1)$	n/a
P1	$\frac{-\pi}{M}[M-(2j-1)][(j-1)M+(i-1)]$	n/a
P2	$\frac{-\pi}{2M}[2i-1-M][2j-1-M]$	n/a
P3	$\frac{\pi}{N_c}(i-1)$	n/a
P4	$\frac{\pi}{N_c}(i-1)^2 - \pi(i-1)$	n/a
T1	$mod\left\{\frac{2\pi}{n}\left\lfloor(kt-jT)\frac{jn}{T}\right\rfloor, 2\pi\right\}$	n/a
T2	$mod\left\{\frac{2\pi}{n}\left\lfloor(kt-jT)\left(\frac{2j-k+1}{T}\right)\frac{n}{2}\right\rfloor, 2\pi\right\}$	n/a
T3	$mod\left\{\frac{2\pi}{n}\left\lfloor\frac{n\Delta Ft^2}{2T}\right\rfloor, 2\pi\right\}$	n/a
T4	$mod\left\{\frac{2\pi}{n}\left\lfloor\frac{n\Delta Ft^2}{2T} - \frac{n\Delta Ft}{2}\right\rfloor, 2\pi\right\}$	n/a

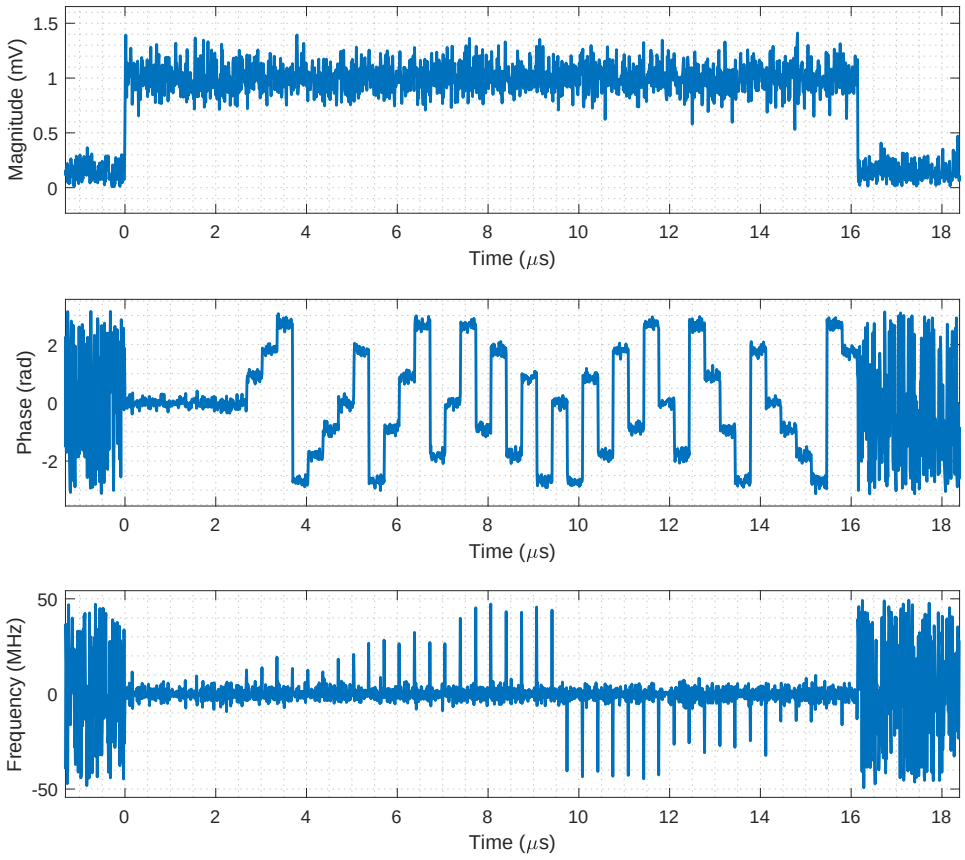


Figure 2.1: Instantaneous magnitude, phase and frequency plots of a generated Frank modulated IMOP data at 15 db SNR.

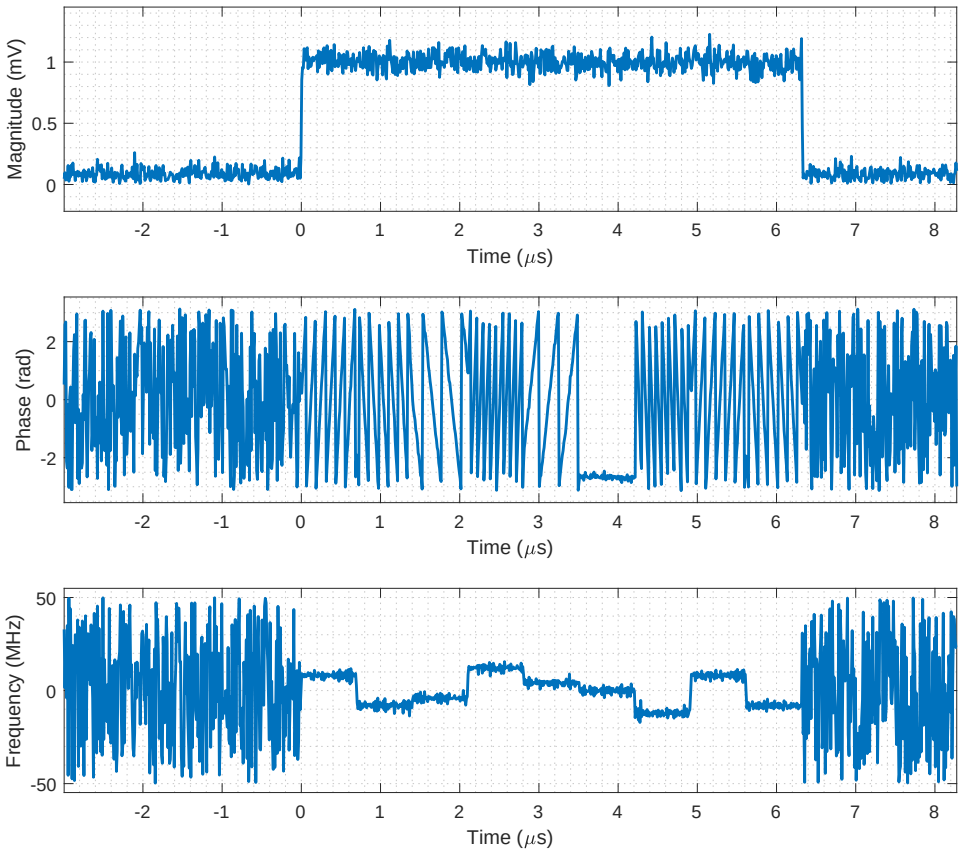


Figure 2.2: Instantaneous magnitude, phase and frequency plots of a generated 5-step frequency modulated (Costas 5) IMOP data at 20 db SNR.

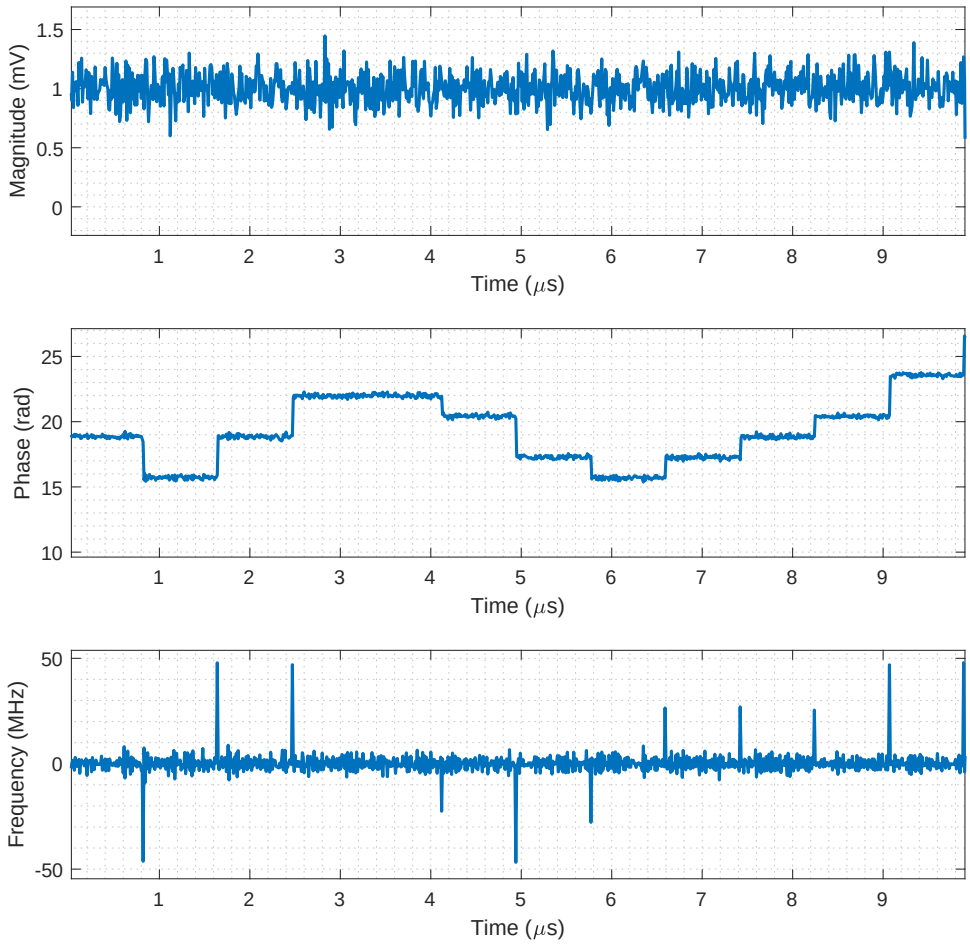


Figure 2.3: Instantaneous magnitude, unwrapped phase and instantaneous frequency plots of a generated QPSK phase modulated IMOP data at 15 db SNR.

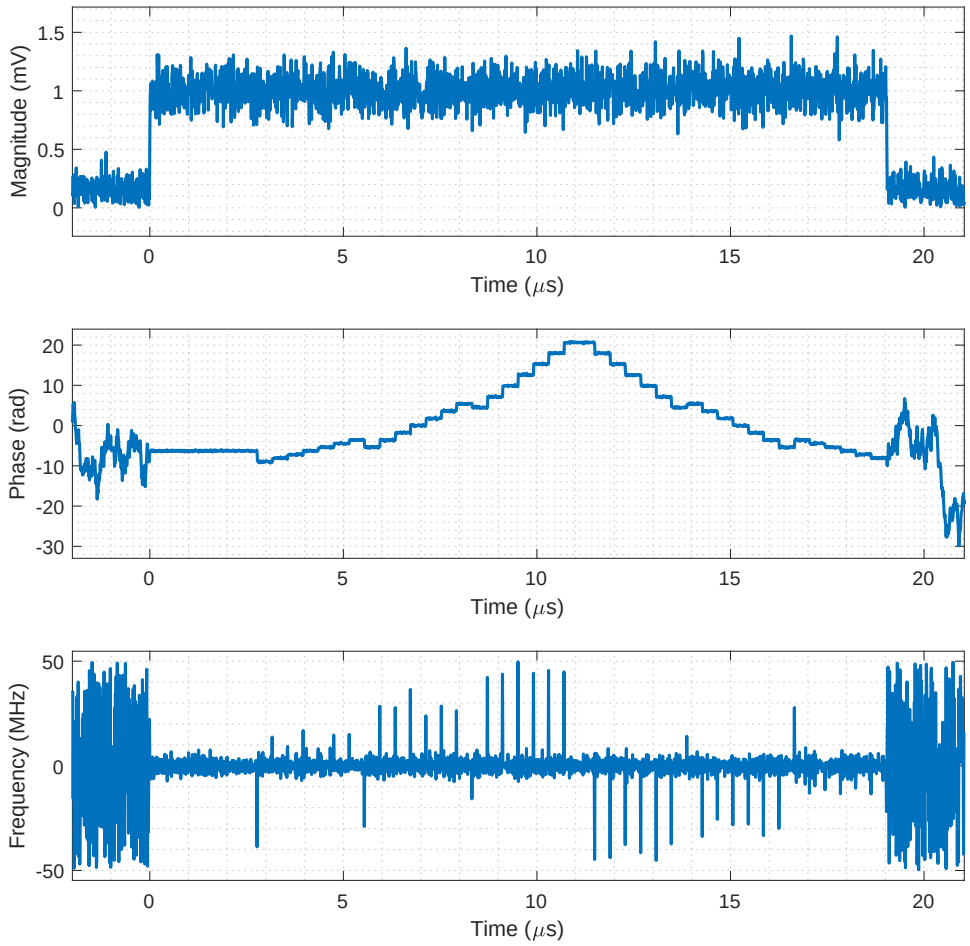


Figure 2.4: Instantaneous magnitude, unwrapped phase and instantaneous frequency plots of a generated P1 polyphase modulated IMOP data at 15 db SNR.

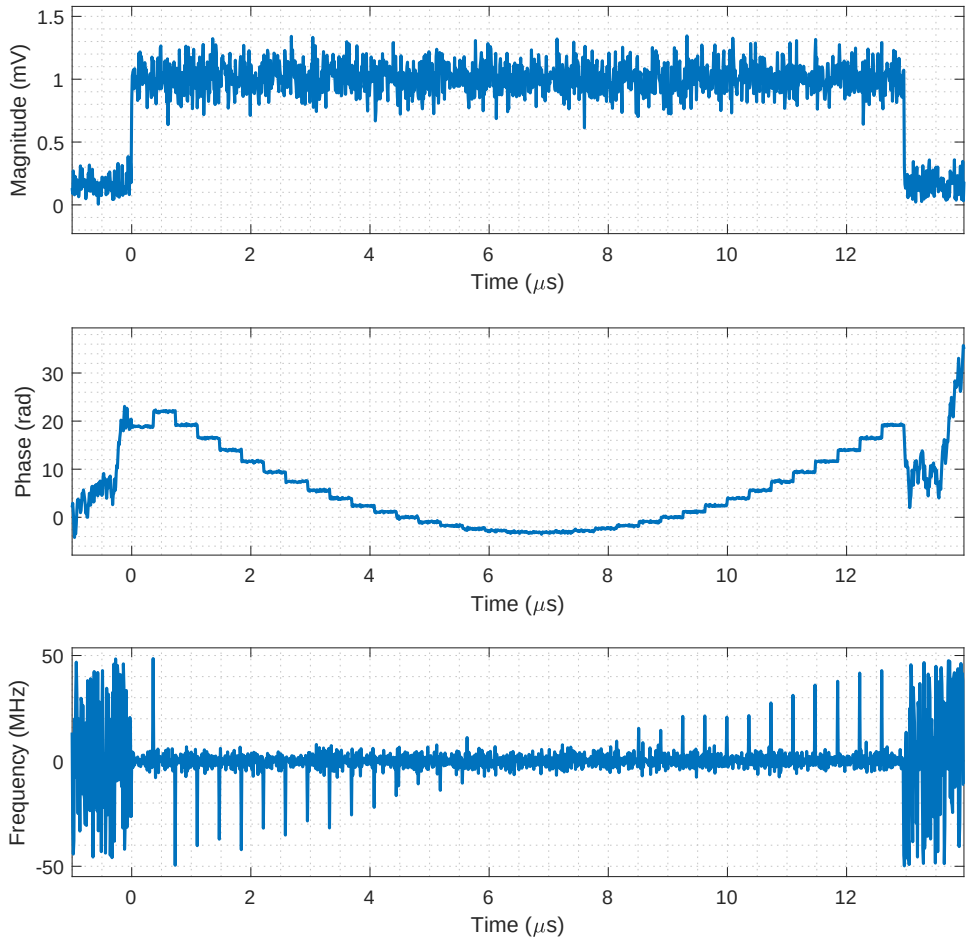


Figure 2.5: Instantaneous magnitude, unwrapped phase and instantaneous frequency plots of a generated P4 polyphase modulated IMOP data at 15 db SNR.

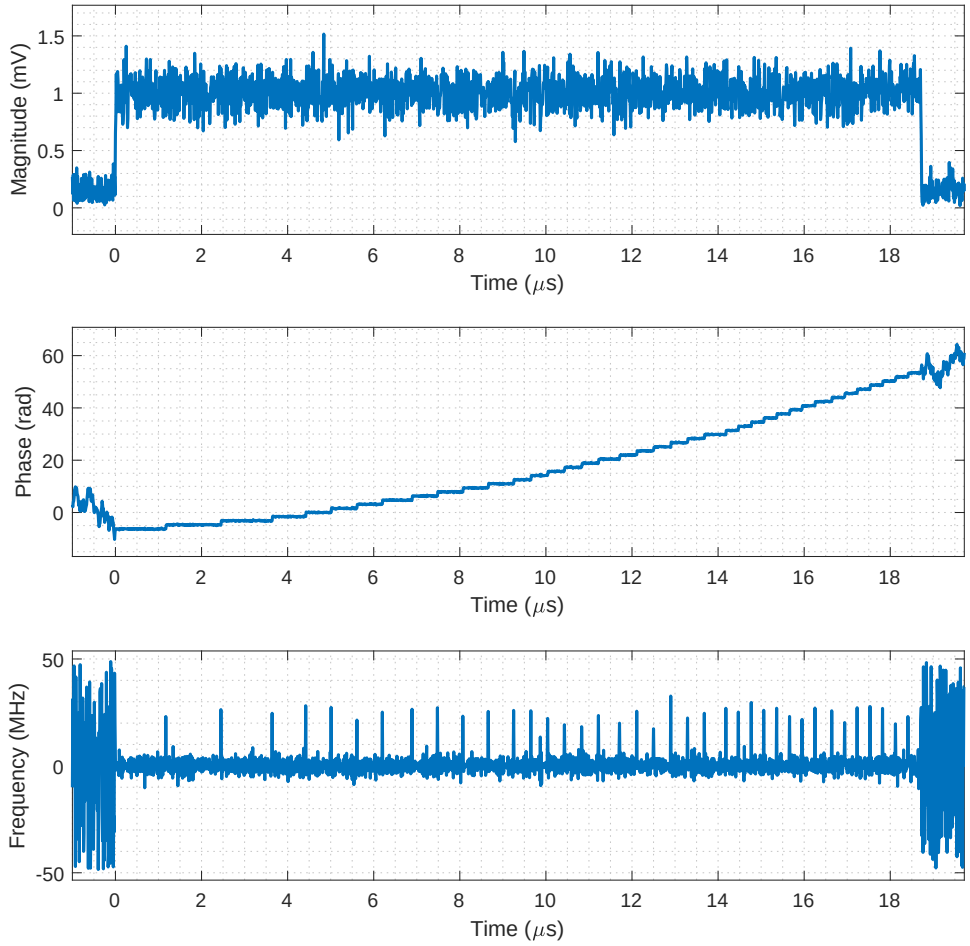


Figure 2.6: Instantaneous magnitude, unwrapped phase and instantaneous frequency plots of a generated T1 polytime phase modulated IMOP data at 15 db SNR.

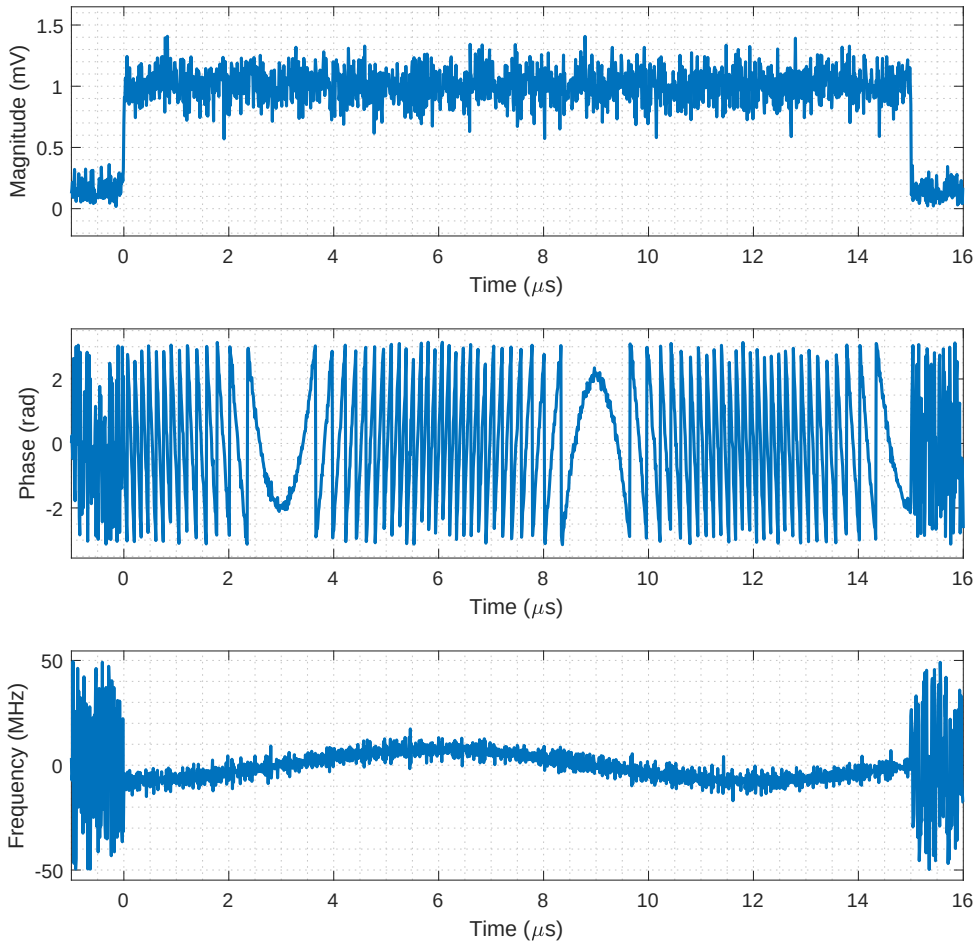


Figure 2.7: Instantaneous magnitude, wrapped phase and instantaneous frequency plots of a generated sinusoidal frequency modulated IMOP data at 15 db SNR.

buttom-left part and plots for a selected IMOP measurement can be seen on the right side of the user-interface. Figure 2.9 shows the section on the user-interface where common parameters of the IMOP signals can be adjusted. After configuring the common parameters and choosing desired modulation type, one can select the details of that modulation type in MOP Parameters section. Figure 2.10 shows this section on the user-interface where modulation specific parameters of the IMOP signals can be adjusted. After adjusting the general and modulation specific parameters, one can add this configured modulation type to the Class List that can be seen in Figure 2.12. Finally, once the Class List is populated with desired modulation parameters data generation can be performed in the section that can be seen in Figure 2.11. In this section one can also set the number of samples per class to be generated, select the type of padding to be used and get time/space estimates on the final data before starting the data generation.

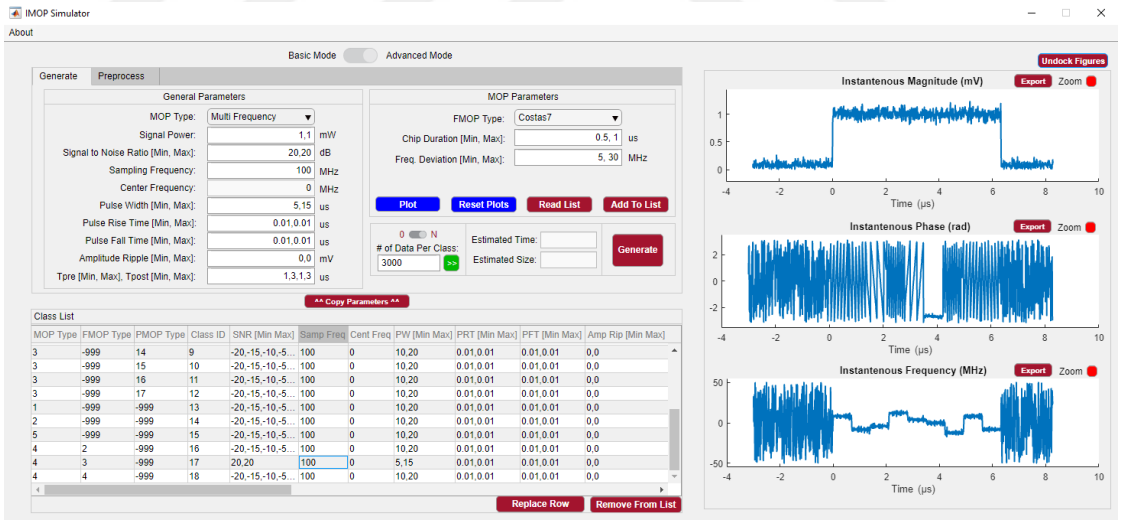


Figure 2.8: Overall screenshot of the IMOP simulator.

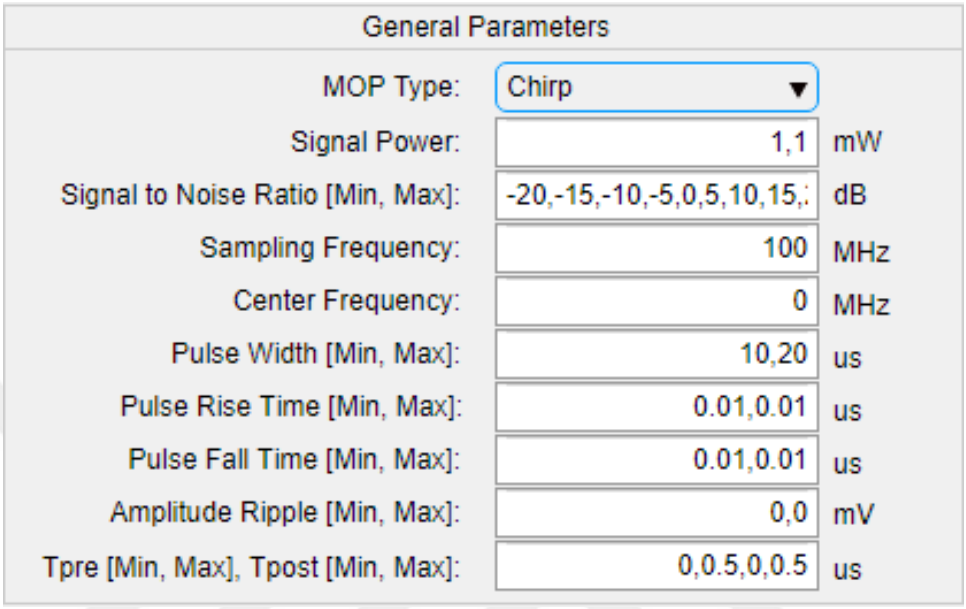


Figure 2.9: General parameters section of the IMOP simulator.

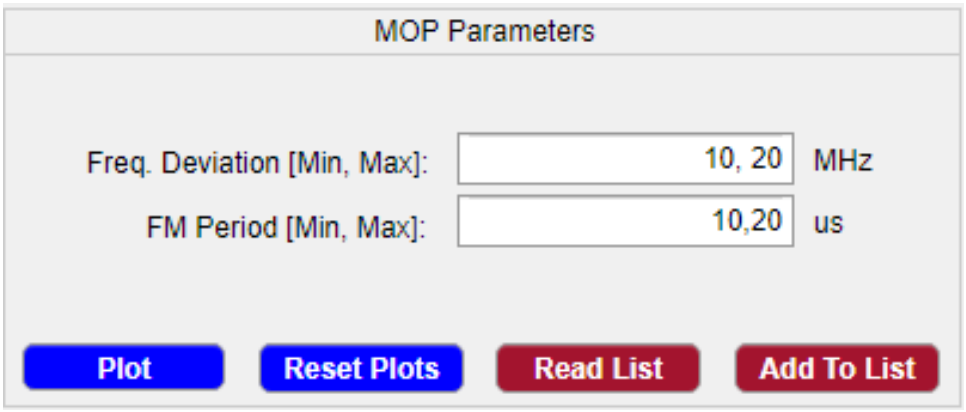


Figure 2.10: MOP parameters section of the IMOP simulator.

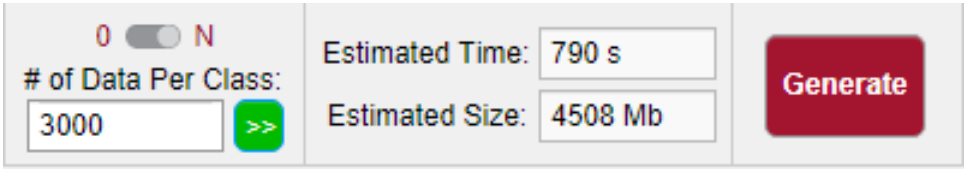


Figure 2.11: Data generation section of the IMOP simulator.

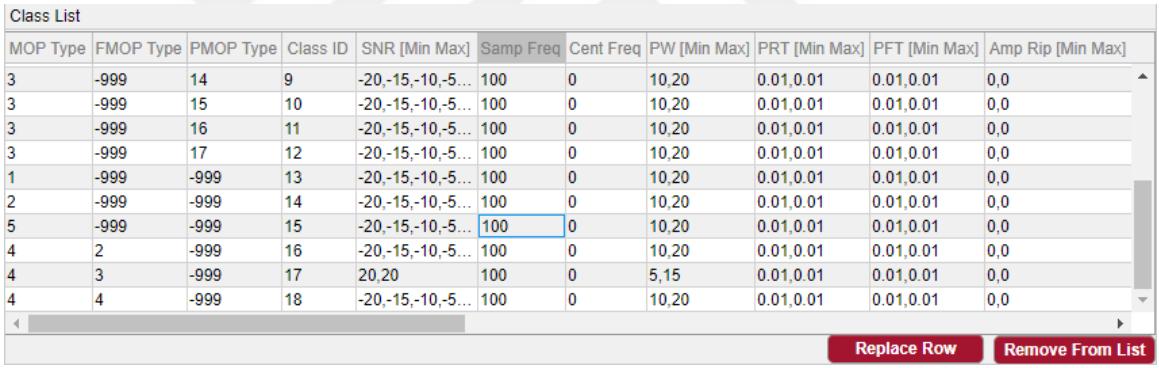


Figure 2.12: Modulation type list section of the IMOP simulator.

Chapter 3

Proposed Methods

3.1 IMOP Recognition by Feature Fusion based Convolutional Neural Network

Two different pre-processing procedures are applied before network in order to facilitate both frequency and phase modulation identification of $x(t)$. This approach is different than traditional learning based methods with handcrafted features, since in FF-CNN these 2 automatically generated inputs are used in a network in an end-to-end manner. In other words, feature learning and classification is performed automatically. First processing extracts Time-Frequency Images (TFIs) of the time-series complex signals which are good for differentiating frequency modulations. However, pseudo-random sequenced phase modulations have very similar TFIs. Thus, the second preprocessing is employed that makes the discrimination of phase modulated signals easier. Below, the pre-processing technique and proposed deep network structure are detailed.

3.1.1 Pre-processing Stages

In the first stage of pre-processing Reassigned Short-Time Fourier Transform (RSTFT) [26] of $x(t)$ is computed to generate high-resolution TFI of $x(t)$ to emphasize frequency modulations. Let $F_x(t, w; z)$ denote the STFT of $x(t)$, given as:

$$F_x(t, w; z) = \int_{-\infty}^{\infty} x(s) z^*(s - t) e^{-jws} ds \tag{3.1}$$

where $z(t)$ is the windowing function controlling the desired time and frequency resolution of the resulting TFI. Then, RSTFT of the detected signal $x(t)$ is computed as:

$$\begin{aligned} S_x^r(t', w') = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} S_x(t, w) \delta(t' - \hat{t}_x(t, w)) \\ \times \delta(w' - \hat{w}_x(t, w)) dt dw \end{aligned} \tag{3.2}$$

where $\hat{t}_x(t, w)$, $\hat{w}_x(t, w)$, and $S_x(t, w)$ are defined as:

$$S_x(t, w) = |F_x(t, w; z)|^2 \tag{3.3}$$

$$\hat{t}_x(t, w) = t - Re \left\{ \frac{F_x(t, w; T_z(t)) F_x^*(t, w; z)}{S_x(t, w)} \right\} \tag{3.4}$$

$$\hat{w}_x(t, w) = w + Im \left\{ \frac{F_x(t, w; D_z(t)) F_x^*(t, w; z)}{S_x(t, w)} \right\} \tag{3.5}$$

with $T_z(t) = tz(t)$ and $D_z(t) = \frac{dz(t)}{dt}$. Fig. 3.4 illustrates the STFT (3.4a) and the RSTFT (3.4b) of a frequency modulated $x(t)$ measured at 10 dB SNR. As demonstrated, the RSTFT provides a higher resolution TFI than STFT. However, since the high resolution TFI's are spatially sparse, they are downsampled to 128$\times$ 128 by the nearest-neighbor interpolation method with a neglegible information loss [27] to train the FF-CNN on a standardized input size with decreased training duration.

In the second stage of pre-preprocessing, the unwrapped instantaneous phase of the measured signal $phase(x(t_i))$ is convolved with $n = 1$ order HG (Hermite Gaussian) given in Eq. 3.6 to form the convolved phase $c_x(t_i)$. In Eq. 3.6, β and σ are amplitude and time parameters, respectively. σ can be chosen so that

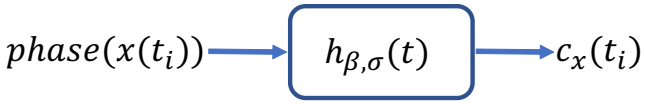


Figure 3.1: In the second stage of pre-preprocessing, the unwrapped instantaneous phase of the i'th measured signal $phase(x(t_i))$ is convolved with $n = 1$ order HG $h_{\beta,\sigma}(t)$ to form the convolved phase $c_x(t_i)$.

Figure 3.2: In the second stage of pre-preprocessing, discontinuities in the convolved phase $c_x(t_i)$ are detected by using Recursive Least Squares (RLS) algorithm and then quantized to form the vector $q_x(b_i)$.

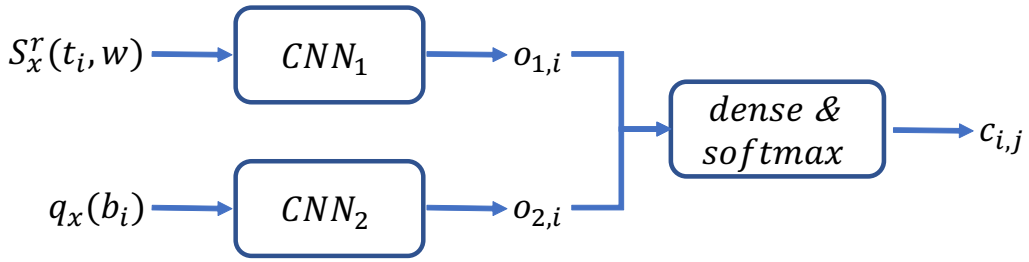
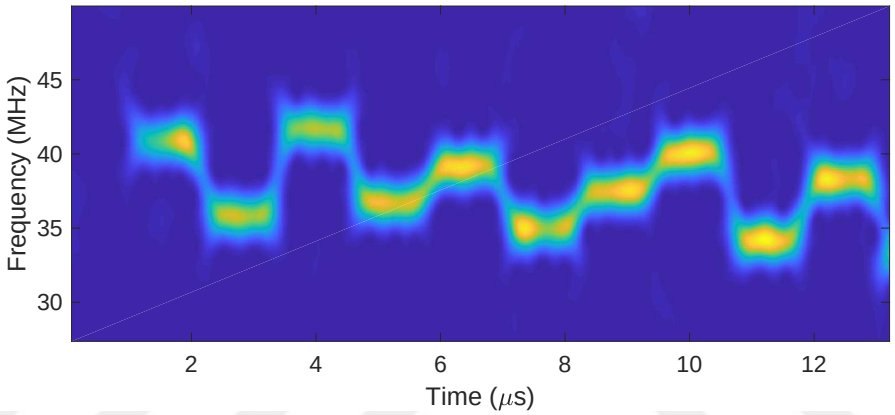
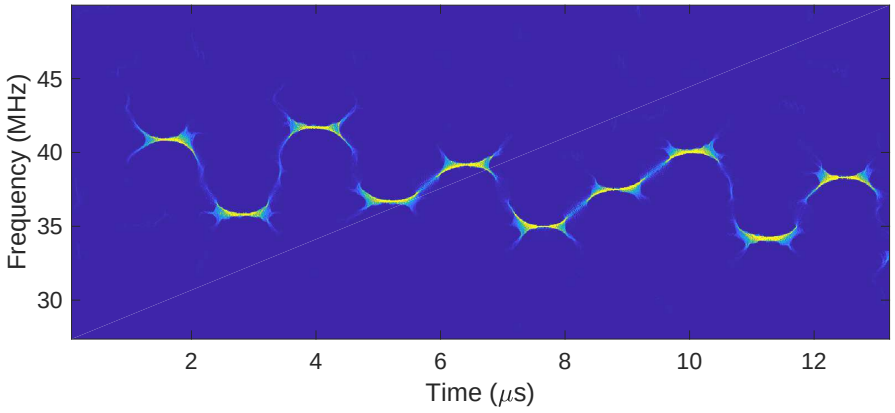


Figure 3.3: The proposed feature fusion based convolutional neural network (FF-CNN) model. First, preprocessed inputs, time-frequency image $S_x^r(t_i, w)$ and quantized phase $q_x(b_i)$, are subjected to feature extraction procedure through two convolutional neural network blocks CNN_1 and CNN_2, then two network outputs $o_{1,i}$ and $o_{2,i}$ are simultaneously fused by concatenation, and finally fed to dense and softmax layers to get the probability vector $c_{i,j}$ which represents the probability that i'th data belongs to the j'th class.

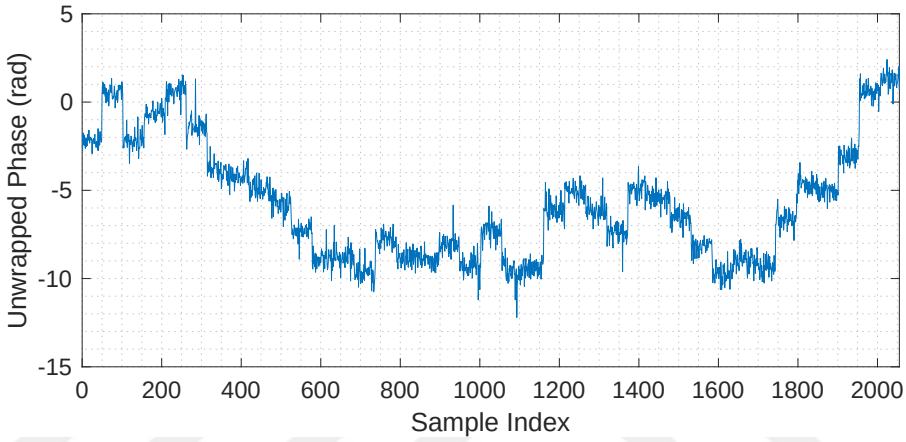


(a)

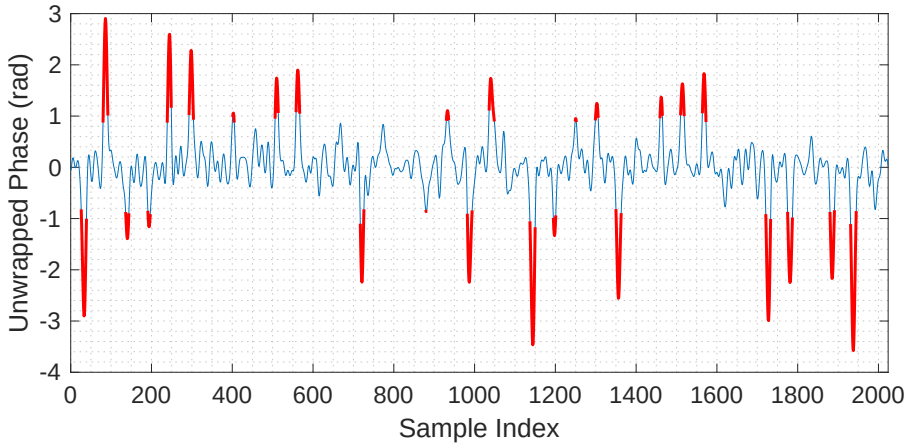


(b)

Figure 3.4: TFI's of a Costas-10 modulated pulse at 10dB SNR using (a) STFT, and (b) RSTFT at 100 MHz sampling frequency.



(a)



(b)

Figure 3.5: Second pre-processing steps for a 16-PSK (phase) modulated pulse at 5 dB SNR. (a) Phase of the modulated signal, detected by applying a threshold. (b) Convolution of the pulse phase with HG (blue), detected phase jumps by robust least squares (red).

effective time support of the $h_{\beta,\sigma}(t_n)$ is set to half of the minimum chip duration. β should be chosen as $\beta = 2/\sum_{-N_h}^{N_h} |h_{1,\sigma}(t_n)|$. Application of this step over i'th measurement is illustrated in Figure 3.1.

Once the convolution is complete, discontinuities in $c_x(t_i)$ are detected robustly by using Recursive Least Squares (RLS), which is one of the most widely used outlier elemination techniques, as given in Algorithm 1.

$$h_{\beta,\sigma}(t_n) = \beta \frac{t_n}{\sigma} e^{\frac{-\pi t^2}{\sigma^2}} \ , n = -N_h, ..., N_h \tag{3.6}$$

Convolution of the detected signal's instantaneous phase with the function $h_{\beta,\sigma}(t_n)$ is equal to effectively smoothed derivation operation, and provides more apparent phase jumps, as illustrated in Fig. 3.5. Outliers of the convolved phase are detected by RLS method [28] and thereby phase shift points are determined, as illustrated in Figure 3.5b. These shift points are then quantized into evenly sized 32 bins between $-\pi$ and $+\pi$ to form the quantized phase vector $q_x(b_i)$. Here b_i represents the bin index for the i'th measurement. This procedure does not provide any output for phase changes in frequency modulated signals.

3.1.2 Convolutional Neural Network Model and Feature Fusion

Convolutional Neural Networks are widely used in image processing related problems for the automatic feature extraction and classification purposes. Input is convolved with a set of filters that of each is specialized for the detection of different local patterns. These convolution filter weights are updated during training phase so that they can detect local similarities in the image better. At the last layer of the CNN, class probabilities are given.

The proposed CNN model can be seen in Fig. 3.3. It takes two inputs time-frequency image $S_x^r(t_i, w)$ and quantized phase $q_x(b_i)$ and returns the modulation

category probabilities $c_{i,j}$ as the output. Reassigned TFI $S_x^r(t_i, w)$ can be obtained by preprocessing of the signal and discretized phase difference vector $q_x(b_i)$ can be determined by and RLS adaptive filter. CNN_1 and CNN_2 architectures given in 3.3 are detailed below.

Frequency modulated signals in time-frequency image form enables recognition through convolutional neural networks as they are in the image form. For the first pre-processed input, feature extraction process is performed in CNN_1 network of three convolutional layers, as illustrated in Fig. 3.3. In these layers, 8, 4 and 2 filters are used with the size of 5x5, 4x4 and 4x4, respectively. The filter sizes are selected so that the lowest local similarity of the TFIs can be learned by the CNN. The unusual pattern with decreasing filter numbers is explained in the last paragraph of this chapter. Max pooling of size 2x2 is performed with stride of 2x2 after each layer to reduce computation, thereby decreasing size.

As for the CNN_2, one-dimensional three layered convolutional neural network is implemented as second feature extraction step using vectors obtained by second pre-processing. In these layers, 8, 4 and 2 filters are used with size of 5x1, 4x1 and 4x1, respectively. Max pooling of size 2x1 with stride of 2x1 is performed after each layer to reduce computation and decrease size.

Lastly, feature fusion is applied to the output vectors of both CNNs $o_{1,i}$ and $o_{2,i}$, by combining latent output vectors $o_{1,i}$ and $o_{2,i}$ of 5 neurons each and passing them through 2 dense layers where classification is performed. When the feature fusion layer is applied to the last layers of the CNNs, and training is performed as a single network instead of two separate classifiers, the resultant network model learns to tolerate errors and weak points of the individual pre-processing methods by adjusting the weights of the extracted features and manages to obtain highly accurate results.

Lastly, in CNNs it is a common approach to increase the number of channels while decreasing layer sizes progressively with the purpose of preventing information loss [29]. However, increasing number of channels also increases the required

computation as well as the number of parameters needs to be learned. In the proposed technique, similar to sparsifying autoencoder structures [30], both the size of layers and the number of channels are decreased to prevent excessive growth in the number of parameters and to ensure reduction in the size of layers progressively. As a result, a CNN structure that can successfully generalize over limited set of training data is obtained.

3.2 End-to-end Pulse Detection and Modulation Recognition by LSTM based Multi-Task Network

A structure based on LSTM and multitasking learning is used to determine the time support of the signal and classify the type of modulation over the measurement $x(t)$. Details of this structure are described in the following subsections.

3.2.1 Long-Short Term Memory Networks

With the development of deep learning recently, ANN has been widely used for processing time series signals. LSTM, a specialized ANN type, has been widely used in time series audio and video signals as it can provide solutions to the gradually decreasing gradients problem in traditional ANNs. [31].

The problem of gradually decreasing gradients can also be considered as an act of forgetting in the human brain. Traditional ANNs have difficulty in connecting information as the signal length increases. LSTM solves this problem by optimizing the transfer of information between memory cells by using its passage structure. An LSTM memory cell is shown in Figure 3.6 and the iteration steps are detailed in [30].

Algorithm 1 Robust Least Squares Estimator with *Bisquare* Weight Function

Inputs: $\mathbf{y} \in \mathcal{R}^K$, N_i, ζ
Outputs: $\mathbf{w} \in \mathcal{R}_+^K$
Initializations: $\mathbf{w} = 1$, $\mathbf{e}^{(0)} = 0$, $\theta = 1$, $i = 1$
while $i \leq N_i$ **do**
  $\mathbf{W} = \text{diag}(\mathbf{w})$
  $\mathbf{b}^{(i)} = \left(\mathbf{1}^T\mathbf{W}\mathbf{1}\right)^{-1}\mathbf{1}^T\mathbf{W}\mathbf{y}$
  $\mathbf{e}^{(i)} = \mathbf{y} - \mathbf{1}\mathbf{b}^{(i)}$
  $\kappa = 6.946\text{med}(\mathbf{e}^{(i)})$
  $\mathbf{w}_k = \begin{cases} 0 & \text{if} |\mathbf{e}_k^{(i)}| > \kappa \\ \left[1 - \left(\mathbf{e}_k^{(i)}/\kappa\right)^2\right]^2 & \text{if} |\mathbf{e}_k^{(i)}| > \kappa \end{cases}$
  if $\|\mathbf{e}^{(i)} - \mathbf{e}^{(i-1)}\|/\|\mathbf{e}^{(i)}\| \leq \zeta$ **then**
    break
  end if
end while

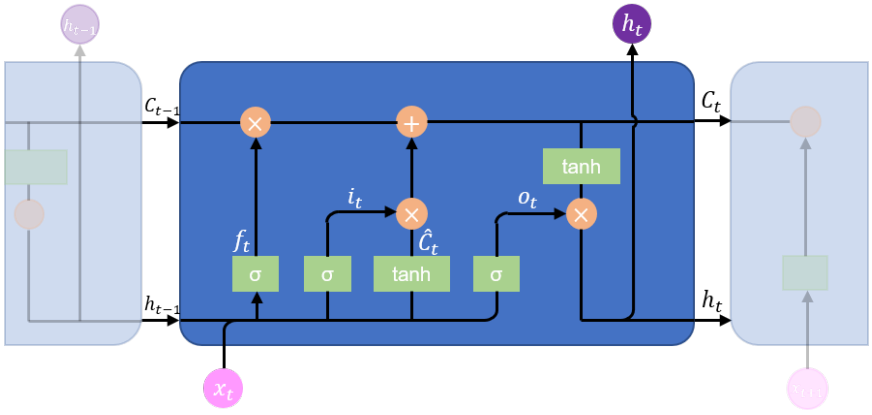


Figure 3.6: Generic illustration of the long short term memory cell structure. Here W is the weight matrix; b is the bias vector; i, f, and o are input, forget and output gates respectively; C and h represents the cell activation and output vector. Lastly, sigmoid and tanh are the nonlinear activation functions.

Recurrent steps of the long short memory networks are given below:

$$f_t = \text{sigmoid}(W_f[h_{t-1}, x_t] + b_f) \tag{3.7}$$

$$i_t = \text{sigmoid}(W_i[h_{t-1}, x_t] + b_i) \tag{3.8}$$

$$\hat{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C) \tag{3.9}$$

$$C_t = f_t C_{t-1} + i_t \hat{C}_t \tag{3.10}$$

$$o_t = \text{sigmoid}(W_o[h_{t-1}, x_t] + b_o) \tag{3.11}$$

$$h_t = o_t \tanh(C_t) \tag{3.12}$$

where W is the weight matrix; b is the bias vector; i, f, and o are input, forget and output gates respectively; C and h represents the cell activation and output vector. Lastly, sigmoid and tanh are the nonlinear activation functions.

3.2.2 Multi Task Learning

Multitasking learning [32] is called the technique of teaching more than one task at the same time to increase the generalization ability of an artificial neural network (can also be considered as applying soft constraints on parameters). When part of the neural network is shared between tasks, the network tends to achieve the better results [30].

In Figure 3.7, generic structure of a multi-task learning based model is illustrated. Here x represents neural network input, and $y^{(i)}$ represents the labels the network tries to learn for different tasks. $h^{(o)}$ shows the shared network parameters learned for tasks, and $h^{(i)}$ shows the task oriented network parameters learned for the related tasks. While the $h^{(o)}$ parameters learn attributes related to the relationship between different tasks, the parameter blocks represented by $h^{(i)}$ only learn the distinctive attributes related to the task they are related to.

3.2.3 MODNET Structure

Architectural details of the proposed LSTM based neural network architecture with two heads is detailed in this section.

First of all, a complex normalization as given in 3.13 is applied to the raw noisy complex measurements $x(t)$ A 2D time series signal $x_n(t)$ is acquired after this normalization stage. This normalization stage for the i'th measurement is illustrated in 3.8.

$$x(t)_{first\_dimension} = \frac{\mathrm{Re}\{x(t)\}}{\max\left(\sqrt{\left(\mathrm{Re}\{x(t)\}\right)^2 + \left(\mathrm{Im}\{x(t)\}\right)^2}\right)} \tag{3.13}$$

$$x(t)_{second\_dimension} = \frac{\mathrm{Im}\{x(t)\}}{\max\left(\sqrt{\left(\mathrm{Re}\{x(t)\}\right)^2 + \left(\mathrm{Im}\{x(t)\}\right)^2}\right)} \tag{3.14}$$

Multitasking learning-based structure in Figure 3.9 has been proposed for pulse detection and modulation classification from the normalized raw noisy IMOP measurements $x_n(t_i)$. First, normalized input $x_n(t_i)$ is subjected to feature extraction procedure through an LSTM block *LSTM Network*$_1$, then latent vector $l(t_i)$, output of *LSTM Network*$_1$, is fed to two separate LSTM blocks *LSTM Network*$_2$ and *LSTM Network*$_3$ to get the detection vector $d(t_i)$ and class probability $c_{i,j}$ respectively. Details of the *LSTM Network*$_1$, *LSTM Network*$_2$ and *LSTM Network*$_3$ are given in Figures 3.10-3.12 respectively and detailed explanations are given in the following paragraphs.

LSTM Network$_2$ provides with the detected time samples and *LSTM Network*$_3$ provides with the predicted modulation type. Detailed structure of the shared backbone (*LSTM Network*$_1$) can be seen in Figure 3.10. Here $l(t_i)$ represents latent vector, coming from *LSTM Network*$_1$, of i'th data in the dataset, $C'_{l,u}$ and $h'_{l,u}$ represents the cell activation and output vector of the u'th LSTM unit in l'th layer respectively. In the experiments, unit number U is used as 32. Outputs of

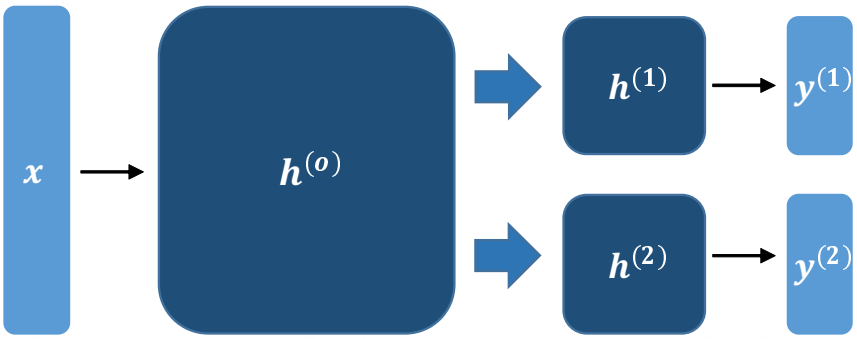


Figure 3.7: Generic illustration of a multi-task learning based model. Here x represents neural network input, and $y^{(i)}$ represents the labels the network tries to learn for different tasks. $h^{(o)}$ shows the shared network parameters learned for tasks, and $h^{(i)}$ shows the task oriented network parameters learned for the related tasks. While the $h^{(o)}$ parameters learn attributes related to the relationship between different tasks, the parameter blocks represented by $h^{(i)}$ only learn the distinctive attributes related to the task they are related to.

Figure 3.8: A complex normalization as given in 3.13 is applied to the raw noisy complex measurements $x(t_i)$. A 2D time series signal $x_n(t_i)$ is acquired after this normalization stage.

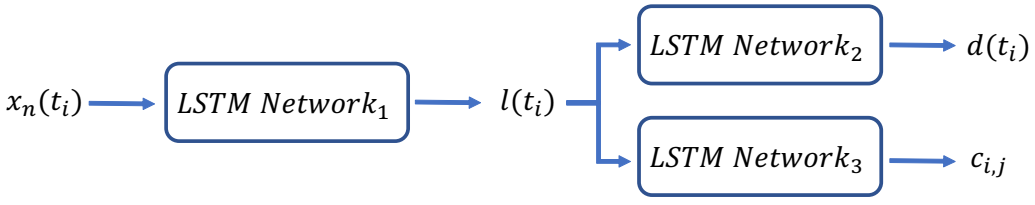


Figure 3.9: The proposed MODNET model. First, normalized input $x_n(t_i)$ is subjected to feature extraction procedure through an LSTM block *LSTM Network*$_1$, then the network outputs $l(t_i)$ is fed to two separate LSTM blocks *LSTM Network*$_2$ and *LSTM Network*$_3$ to get the detection vector $d(t_i)$ and class probability $c_{i,j}$ respectively.

the given LSTM network is the detection vector $d(t_i)$ corresponding the probability of detection for each time sample of the input signal $x(t_i)$. Figure 3.11 shows the details of the *LSTM Network*$_2$. Here $l(t_i)$ represents latent vector, coming from *LSTM Network*$_1$, of i'th data in the dataset, $C'_{l,u}$ and $h'_{l,u}$ represents the cell activation and output vector of the u'th LSTM unit in l'th layer respectively. Output of the given LSTM network is the detection vector $d(t_i)$ corresponding the probability of detection for each time sample of the input signal $x(t_i)$.

Figure 3.12 shows the details of the *LSTM Network*$_3$. Here $l(t_i)$ represents latent vector, coming from *LSTM Network*$_1$, of i'th data in the dataset, $C''_{l,u}$ and $h''_{l,u}$ represents the cell activation and output vector of the u'th LSTM unit in l'th layer respectively. $d_{u,j}$ represents the dense layer weight between u'th LSTM unit in in the last layer of the decoder network and j'th unit of the output layer $(c_{i,j})$. Output of the given LSTM network is the classification vector $c_{i,j}$ which represents the probability that i'th data belongs to the j'th class.

During the neural network training, the error at each step is calculated from the predictions $\hat{c}$ and $\hat{d}(t)$ of outputs c and $d(t)$, respectively as follows:

$$L_1 = \frac{1}{T}\sum_{i=1}^{T}\left[-\frac{1}{N}\sum_{j=1}^{N}\left[d(t_{ij})\log(\hat{d}(t_{ij}))\right]\right] + \frac{1}{T}\sum_{i=1}^{T}\left[-\frac{1}{N}\sum_{j=1}^{N}\left[(1-d(t_{ij}))\log(1-\hat{d}(t_{ij}))\right]\right] \tag{3.15}$$

$$L_2 = -\frac{1}{T}\sum_{i=1}^{T}\left[c_i\log(\hat{c}_i)\right] \tag{3.16}$$

$$L_T = L_1 + \lambda L_2 \tag{3.17}$$

Here $d(t_{ij})$ and $\hat{d}(t_{ij})$ represent the label and predicted detection result of the j'th element of the i'th data from dataset, respectively. c_i and $\hat{c}_i$ represents the label and predicted modulation type of the i'th measurement; N represents the number of samples in a selected data and T represent the number of data selected from dataset. The problem of estimating $d(t_i)$ is modeled as a binary

classification problem (signal is present or not) for each sample in a data, and optimization with the binary cross-disorder error function. The neural network's parameters are updated with the back propagation algorithm to reduce the total error expressed by L_T and the details of the optimization steps are given in 3.2.5.

At the end of the offline supervised training, the memory cells in shared backbone (*LSTM Network*$_1$) learn the relationship between the modulation type and the pulse detection tasks. Meanwhile, it is provided to learn task-specific attributes with the *LSTM Network*$_2$ and *LSTM Network*$_3$.

3.2.4 Time Analysis

In order to show the applicability of the proposed MODNET architecture in real-world scenarios, we show the time-complexity of the detection output for inference. All calculations are done considering the parallel computation capabilities of the GPUs [30]. Consider an end-to-end network consisting of *LSTM Network*$_1$ and *LSTM Network*$_3$. Although most of the calculations can be parallelized, the calculations in series LSTM units have to be performed in separate cycles [33]. The output can be read from the LSTM after a number of time steps that is asymptotically linear in the number of units and layers [30]. Each LSTM unit has 3 multiplication, 1 addition, 3 sigmoid and 2 tanh activation functions as can be seen in figure 3.6. Time complexity of the activation functions can be considered as constant since they can be utilized by look-up tables [34], thus we can safely ignore them in the calculation. Moreover, the number of clock cycles is the dominant term for time complexity considering the FLOPS of the GPU is not reached per cycle, so we can also ignore the number of multiplications and additions. Then, considering there are L number of layers and each layer has U number of LSTM units, total number of cycles per input sample become $U+L-1$. Ultimately, worst-case time complexity can be expressed as $O\{U+L\}$. With $U = 32$ and $L = 7$ as used in our work and *Tesla V100* GPU with 1370 MHz clock speed, maximum sampling frequency for the receiver is calculated as 17.5 MHz (assuming multiplication is done in 2 cycles with CUDA) for real time

application of the MODNET. In order to achieve 100 MHz sampling frequency, GPU with a 7800 MHz clock speed would be required which could be possible in the foreseeable future.

3.2.5 Optimization

Gradient descent based optimization techniques are used during the supervised training of the neural network based architectures. Update steps of the optimization methods that are utilized in the training of the proposed methods are detailed in this subsection.

Loss minimization problem with stochastic gradient descent can be formulated as follows:

$$\min_{W} \left\{ L_t(W) := \frac{1}{b} \sum\nolimits_{j=1}^{b} \ell(W; x_{i_j}, y_{i_j}) + \lambda r(W) \right\} \tag{3.18}$$

where x_i is the i_{th} training instances and y_i is the corresponding label, W is the network parameters to learn, $\ell(W; x_i, y_i)$ is the loss of network parameterized by W w.r.t. (x_i, y_i), $r(W)$ is the regularization function, $\lambda > 0$ is the regularization weight and $\{(x_{i_j}, y_{i_j})\}_{j=1}^{b}$ corresponds to random *mini-batch* chosen at iteration t.

Weight update rule at t_{th} iteration is formulated as:

$$V_t = \mu V_{t-1} - \alpha \nabla L_t(W_{t-1}) \tag{3.19}$$

$$W_t = W_{t-1} + V_t \tag{3.20}$$

where $\alpha > 0$ and $\mu \in [0, 1)$ corresponds to *learning rate* and *momentum*, respectively. The addition of momentum term smooths updates, enhancing stability and speed.

In Adaptive Gradient (AdaGrad) [35] learning rate is adapted per coordinate in a way that highly varying coordinate suppress and rarely varying coordinate

enhance. Its update rule at t_{th} iteration is formulated as:

$$W_t = W_{t-1} - \alpha \frac{\nabla L_t(W_{t-1})}{\sqrt{\sum_{t'=1}^{t} \nabla L_{t'}(W_{t'-1})^2}} \tag{3.21}$$

Root Mean Square Propagation (RMSProp) [36] is similar to AdaGrad but with an exponential moving average controlled by $\gamma \in [0, 1)$ (smaller $\gamma \implies$ more emphasis on recent gradients). Its weight update rule at t_{th} iteration is formulated as:

$$R_t = \gamma R_{t-1} + (1 - \gamma) \nabla L_t(W_{t-1})^2 \tag{3.22}$$

$$W_t = W_{t-1} - \alpha \frac{\nabla L_t(W_{t-1})}{\sqrt{R_t}} \tag{3.23}$$

Adaptive Moment Estimation (Adam) [37] combines the advantages of AdaGrad, that works well with sparse gradients, and RMSProp, that works well in non-stationary settings. It maintains exponential moving averages of gradient and its square and updates proportional to $\frac{\text{average gradient}}{\sqrt{\text{average squared gradient}}}$. Adam update steps are detailed in Algorithm 2. In the given pseudo code $\alpha > 0$, $\beta_1 \in [0, 1)$, $\beta_2 \in [0, 1)$ and $\epsilon > 0$ are corresponding to learning rate, 1st moment decay rate, 2nd moment decay rate and numerical term, respectively.

In our works, Adam parameters α, β_1, β_2 and ϵ are chosen as 0.001, 0.9, 0.999 and 10^{-8}. Moreover, the regularization constant, which is given as λ in Equation 3.17, is selected as 1.

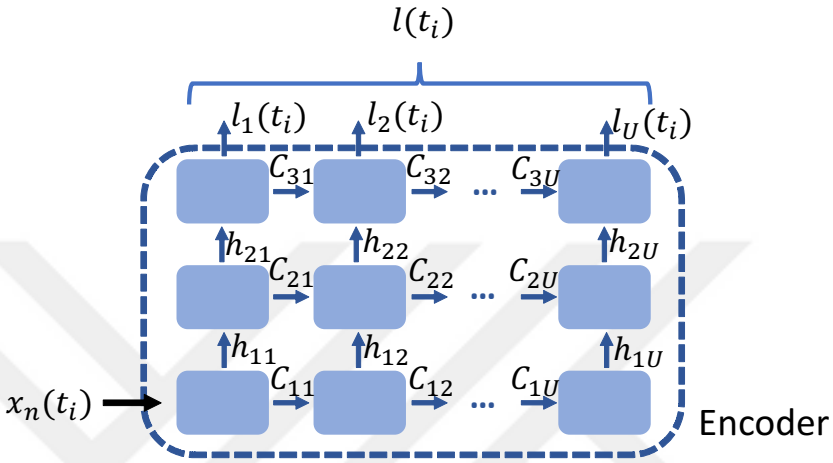


Figure 3.10: Detailed structure of the shared backbone (*LSTM Network*$_1$). Here each blue cell represents the generic LSTM unit given in Figure 3.6, $x_n(t_i)$ represents the normalized samples of i'th data in the dataset, $C_{l,u}$ and $h_{l,u}$ represents the cell activation and output vector of the u'th LSTM unit in l'th layer respectively. Output of the given LSTM network is the latent vector $l(t_i)$ which is the encoded version of the input signal $x_n(t_i)$.

Algorithm 2 Adaptive Moment Estimation (Adam)

Initializations: $M_0 = 0, R_0 = 0$
for $t = 1, \ldots, T$ **do**
  $M_t = \beta_1 M_{t-1} + (1-\beta_1)\nabla L_t(W_{t-1})$ (1st moment estimate)
  $R_t = \beta_2 R_{t-1} + (1-\beta_2)\nabla L_t(W_{t-1})^2$ (2nd moment estimate)
  $\hat{M}_t = M_t / (1-(\beta_1)^t)$ (1st moment bias correction)
  $\hat{R}_t = R_t / (1-(\beta_2)^t)$ (2nd moment bias correction)
  $W_t = W_{t-1} - \alpha \frac{\hat{M}_t}{\sqrt{\hat{R}_t + \epsilon}}$ (Update)
end for
Return W_T

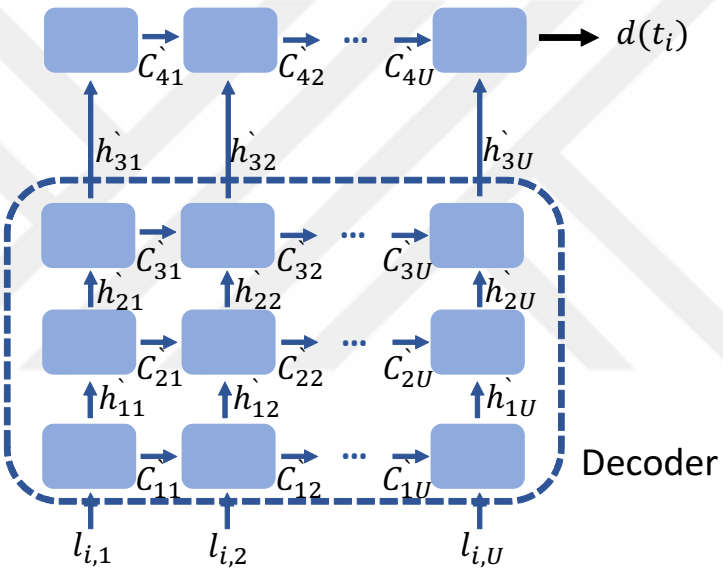


Figure 3.11: Detailed structure of the detection head (*LSTM Network*$_2$). Here each blue cell represents the generic LSTM unit given in Figure 3.6, $l(t_i)$ represents latent vector, coming from *LSTM Network*$_1$, of i'th data in the dataset, $C'_{l,u}$ and $h'_{l,u}$ represents the cell activation and output vector of the u'th LSTM unit in l'th layer respectively. Output of the given LSTM network is the detection vector $d(t_i)$ corresponding the probability of detection for each time sample of the input signal $x(t_i)$.

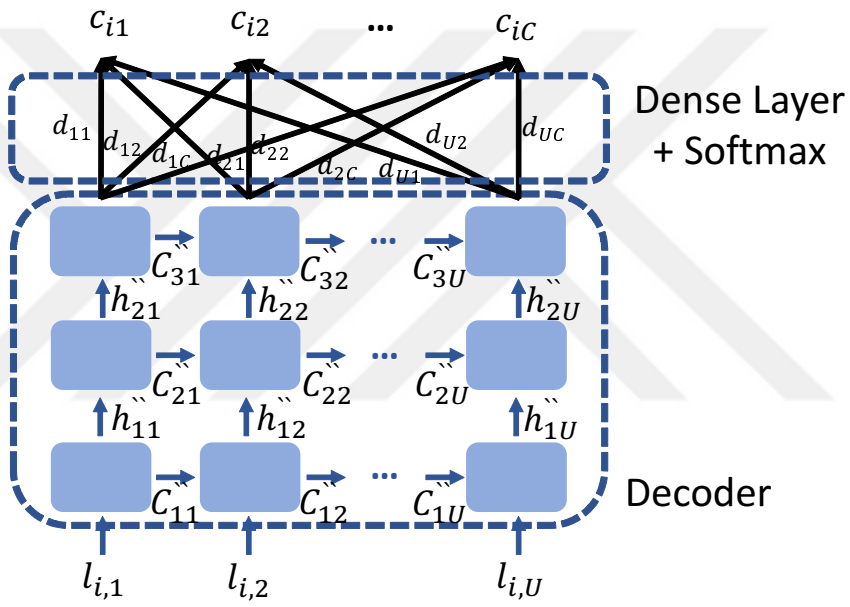


Figure 3.12: Detailed structure of the classification head (*LSTM Network*$_3$). Here each blue cell represents the generic LSTM unit given in Figure 3.6, $l(t_i)$ represents latent vector, coming from *LSTM Network*$_1$, of i'th data in the dataset, $C''_{l,u}$ and $h''_{l,u}$ represents the cell activation and output vector of the u'th LSTM unit in l'th layer respectively. $d_{u,j}$ represents the dense layer weight between u'th LSTM unit in in the last layer of the decoder network and j'th unit of the output layer ($c_{i,j}$). Output of the given LSTM network is the classification vector $c_{i,j}$ which represents the probability that i'th data belongs to the j'th class.

Chapter 4

Experiments and Results

In this chapter, we present the experiments we conducted by training the neural network based models in single-task and multi-task learning setups and applying statistical detection methods. Proposed IMOP data generator was used to form the necessary datasets for training and evaluating the models.

4.1 Datasets

In both modulation classification and pulse detection tasks, synthetic pulses with varying PW values from $(2 - 25)$ μs are generated at 100 Mhz sampling rate from -20 to 20 dB SNR levels. Chosen values for SNR provide challenging test cases at very low SNR values. Signals with changing power levels between 1 mW and 1 W are generated so that pulses that has periodic frequency modulations (ramp, triangular and sinusoidal FM) have at least one period is present in synthetic measurements. Stepped modulations are generated with at least 0.4 μs chip duration. Center frequency of the signals are varied between -5 and 5 MHz. Number of frequency steps for Frank, P1 and P2 coded pulses are selected uniformly from $\{6, 7, 8\}$, and number of sub code in a code is selected uniformly from $\{36, 49, 64\}$ for P3 and P4 coded pulses. Number of segments are chosen

uniformly from $\{4, 5, 6\}$ for T1 an T2 codes, and bandwidth of the intercepted signals are uniformly selected from $(5, 10)$ Mhz for linear, ramp, sinusoidal FM, and T3-T4 coded pulses.

For the overall comparison of the classification and detection models, a dataset of 6 classes, given in Table 4.1, has been formed which contains the widely used frequency and phase modulations. More comprehensive set of 19 classes, given in Table 4.1, is then used to further analyze the proposed detection and classification techniques.

4.2 Experiments

All neural network based models are implemented in in Python using Tensorflow library (v2.1) and optimized with Adam solver [37], which combines the benefits of RMSProp and AdaGrad techniques. SincNet model is adapted from the original repository of [15], autoencoder based detector is implemented as a simple feedforward neural network with 128-64-32-32-64-128 neurons on layers. All measurements are divided into blocks of 2500 time samples. Then 5000 of these blocks were used for training purposes and 3000 of them were used for testing purposes per class, and 10 % of the training set was used as validation data. Training is performed in batches of 64. All of these training, validation and test sets are chosen as mutually exclusive, in other words, network is tested on a set that it has not seen during training phase. Training is performed three times per scenario, and the results are averaged to calculate the classification accuracy.

4.2.1 IMOP Classification Results

Comparison of the classification models over varying SNR levels can be seen in Figure 4.1. Here, CNN corresponds to simple convolutional neural network similar to the one proposed in [10]. WVTFI and CWTFI corresponds to Wigner-Ville and Choi-Williams based feature extractors combined with a CNN model similar

Table 4.1: Modulation Types Used in Simulation Result Sets

6 Class Set	**19 Class Set**	
Single Car. Mod. (SCM)	SCM	Frank Code
Linear FM	Linear FM	P1 Code
Costas-10 FM	Sinusoidal FM	P2 Code
Barker-13 PM	Costas-5 FM	P3 Code
QPSK	Costas-7 FM	P4 Code
8-PSK	Costas-10 FM	T1 Code
	Barker-3 PM	T2 Code
	Barker-7 PM	T3 Code
	QPSK	T4 Code
	8-PSK	

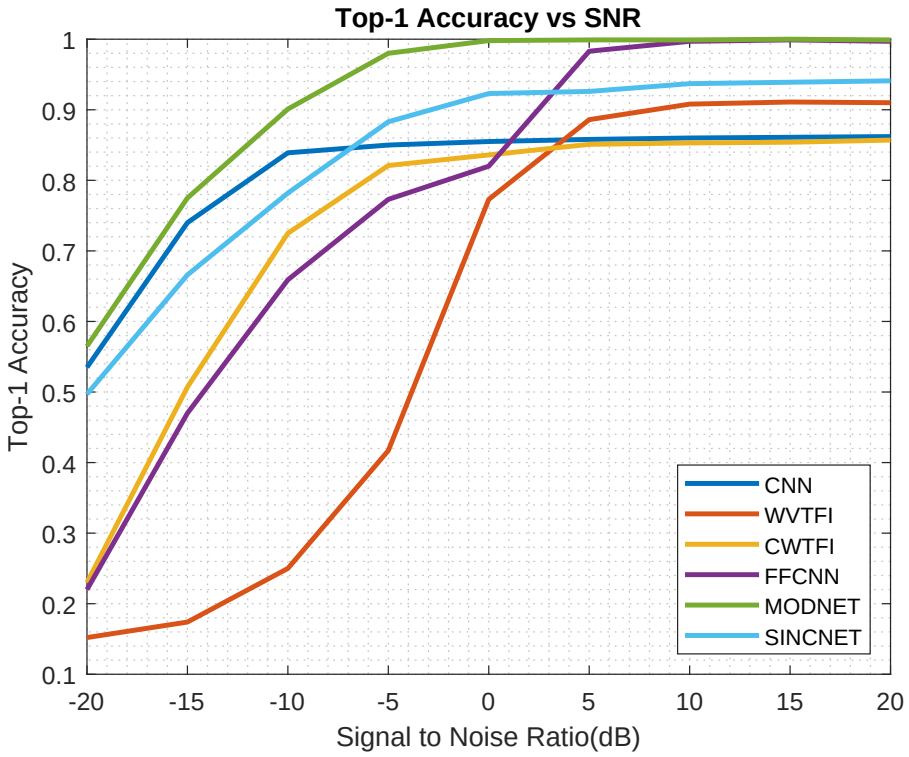


Figure 4.1: Top-1 IMOP classification accuracies over varying SNR levels.

to the one proposed in [2]. SincNet corresponds to the SincNet based IMOP classification model described in Section 5.1.2. FFCNN is the the proposed reassigned spectrum, robust least squares and CNN based IMOP classification technique described in Section 3.1. MODNET is the proposed LSTM and multi-task learning based IMOP classification technique described in 3.2. While calculating the classification accuracy of it, only the output from classification head is regarded during the inference.

As can be seen in Figure 4.1, FFCNN and MODNET are the top performing techniques on 0-20 db SNR region by a large margin. Below 0 db SNR, phase of the measurements start to get distorted largely. As a result, FFCNN's phase related input becomes obsolete and FFCNN fails to differentiate phase modulations (reason of the sudden drop in accuracy). WV and CW based TFI representations are great for frequency related modulations but they are simply not enough to differentiate phase modulations (QPSK and 8-PSK). Moreover, WV TFI representation fails to extract meaningful features to differentiate frequency modulations below 0-5 dB SNR. Overall, MODNET performs better at all SNR levels.

Confusion matrix plots at different SNR levels for IMOP classification results of the proposed FFCNN architecture on 6 class set can be seen in Figures 4.4 and 4.5. At above 0 dB SNR, top accuracies can be observed with near zero false positive and false negative ratios. At 0 dB SNR it is observed that phase modulated classes are misclassified. This is expected since around 0 dB SNR, the quantized phase features become insufficient for the differentiation of the phase jumps.

Confusion matrix plots at different SNR levels for IMOP classification results of the proposed MODNET architecture on 19 class set can be seen in Figures 4.2 and 4.3. Most of the incorrect classifications are between QPSK-8PSK P1-T4, and Chirp-Sin-Costas groups, in other words, there is near to none incorrect classifications between phase-frequency modulations. This is very promising result for low SNR levels.

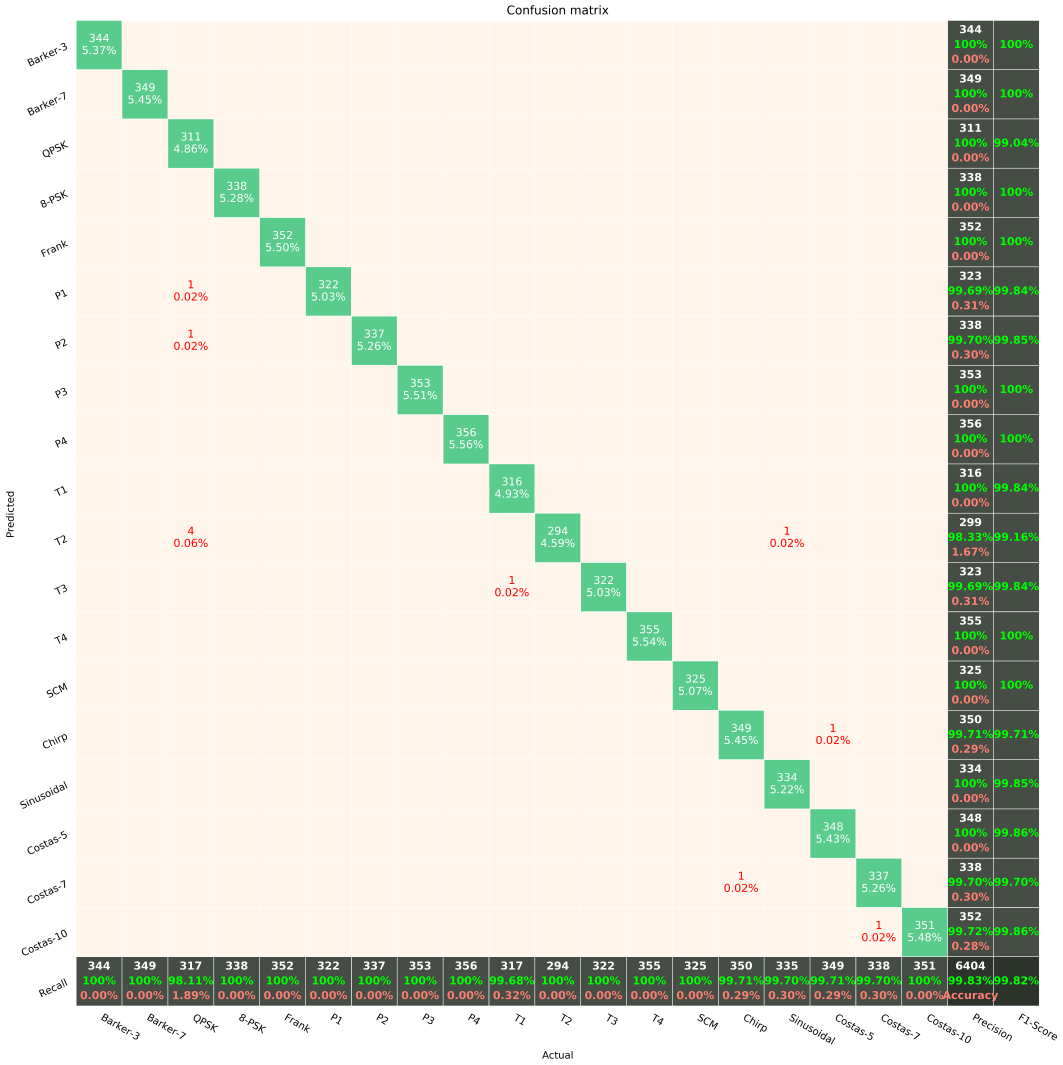


Figure 4.2: Confusion matrix of the proposed ModNet classification head output for test data of 19 class set at 10 dB SNR level.

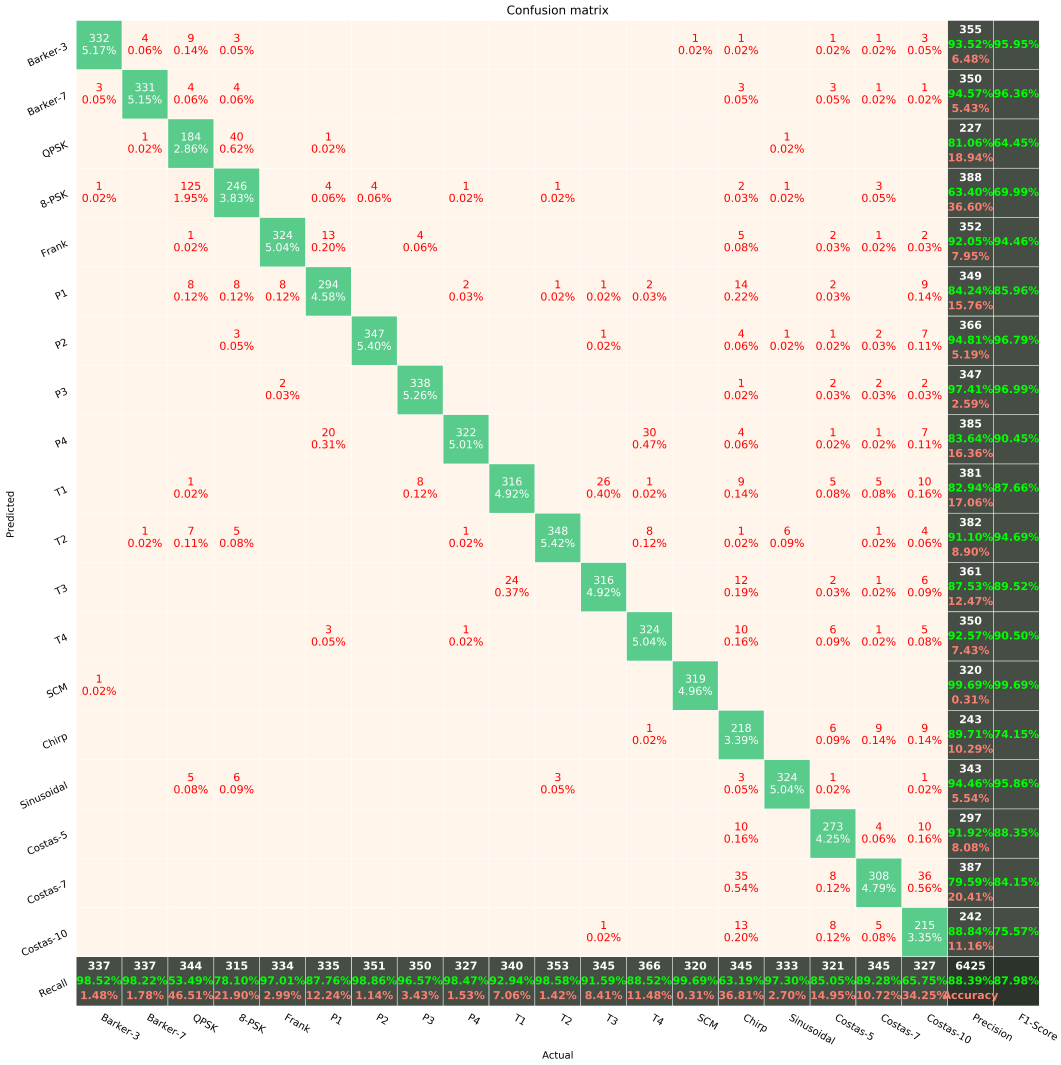


Figure 4.3: Confusion matrix of the proposed ModNet classification head output for test data of 19 class set at -10 dB SNR level.

Figure 4.4: Confusion matrix of the proposed FFCNN technique for test data of 6 class set at (a) 10 dB, and (b) 0 dB SNR levels.

Figure 4.5: Confusion matrix of the proposed FFCNN technique for test data of 6 class set at (a) 10 dB, and (b) 0 dB SNR levels.

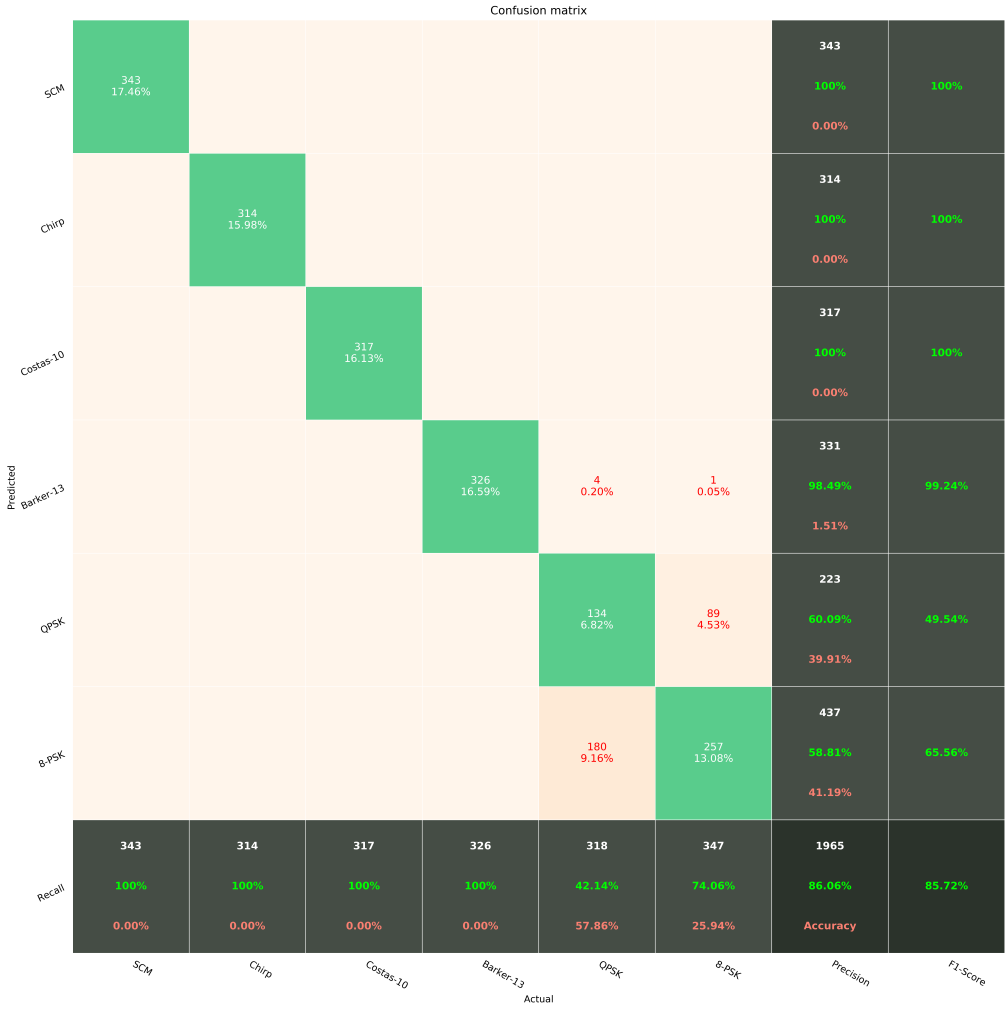


Figure 4.6: Confusion matrix of the pure CNN based model for test data of 6 class set at 10 dB SNR level.

Figure 4.7: Confusion matrix of the pure CNN based model for test data of 6 class set at -10 dB SNR level.

Figure 4.8: Confusion matrix of the WVTFI technique for test data of 6 class set at 10 dB SNR level.

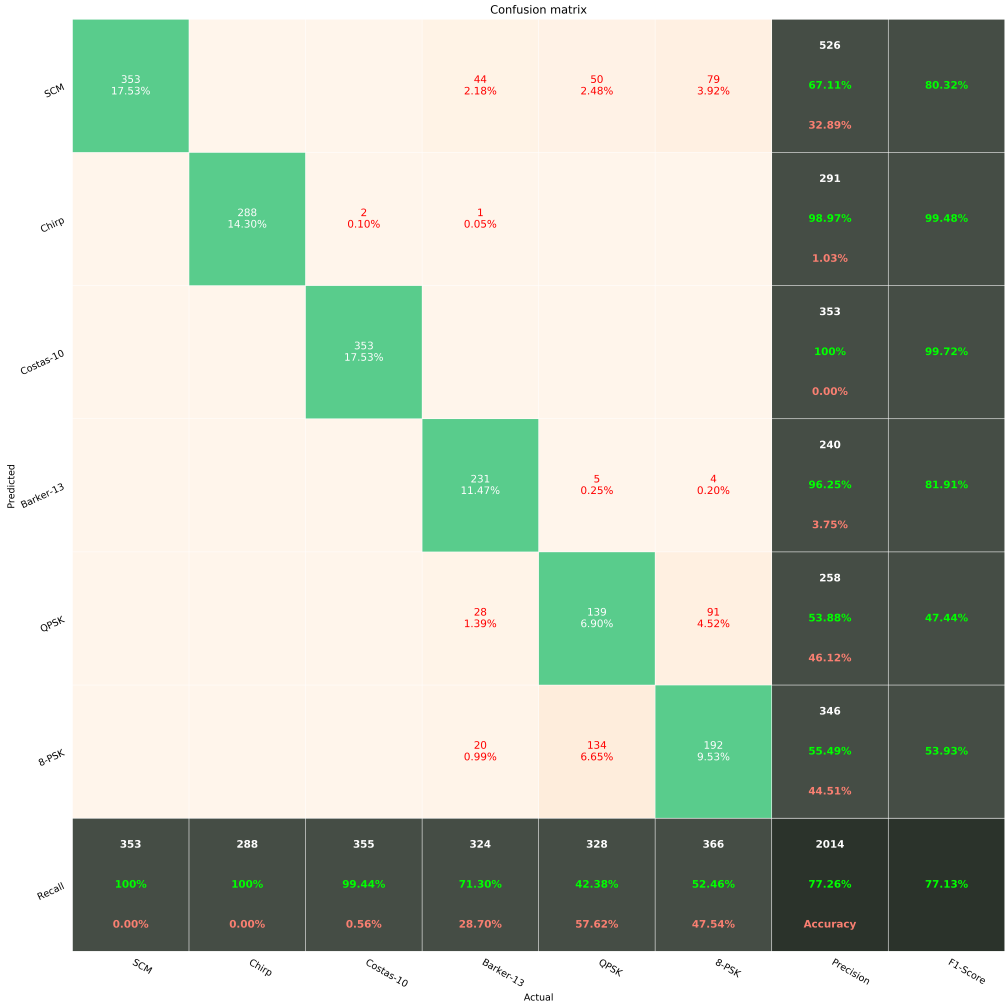


Figure 4.9: Confusion matrix of the WVTFI technique for test data of 6 class set at 0 dB SNR level.

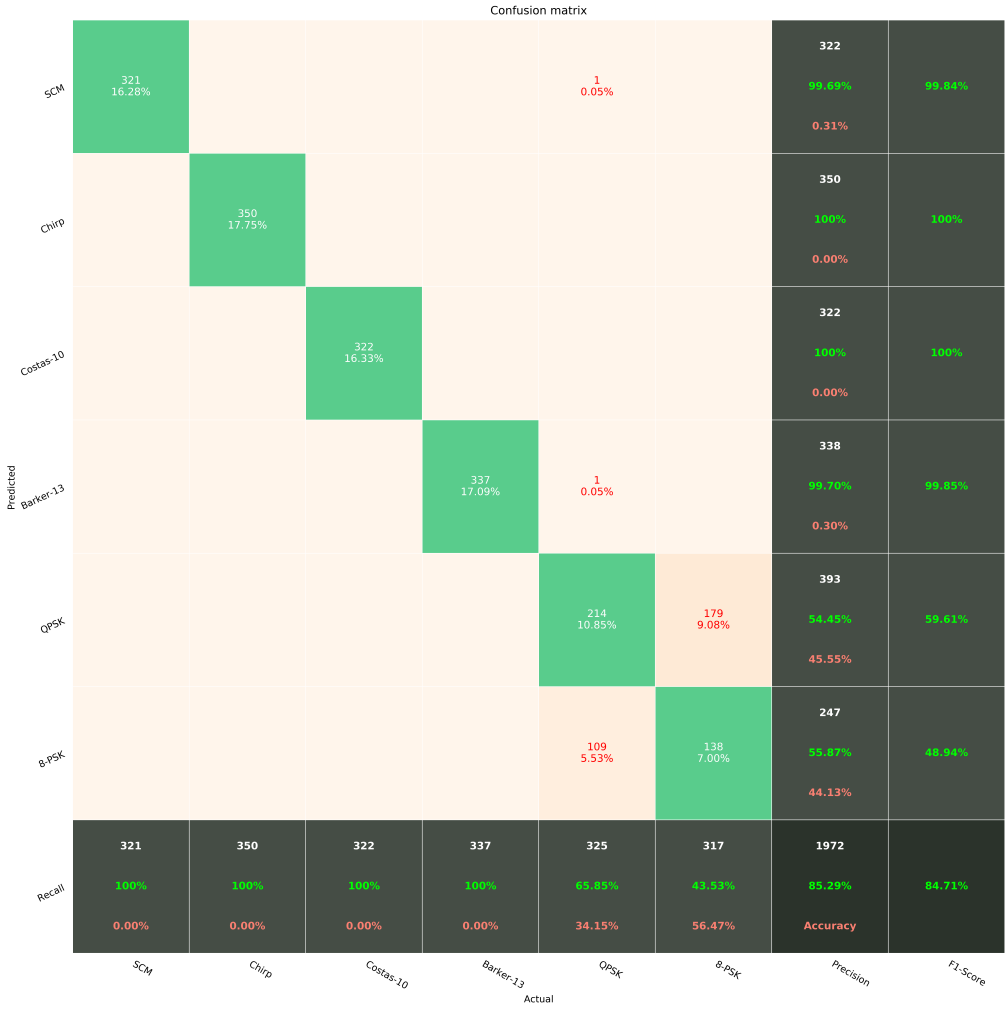


Figure 4.10: Confusion matrix of the CWTFI technique for test data of 6 class set at 10 dB SNR level.

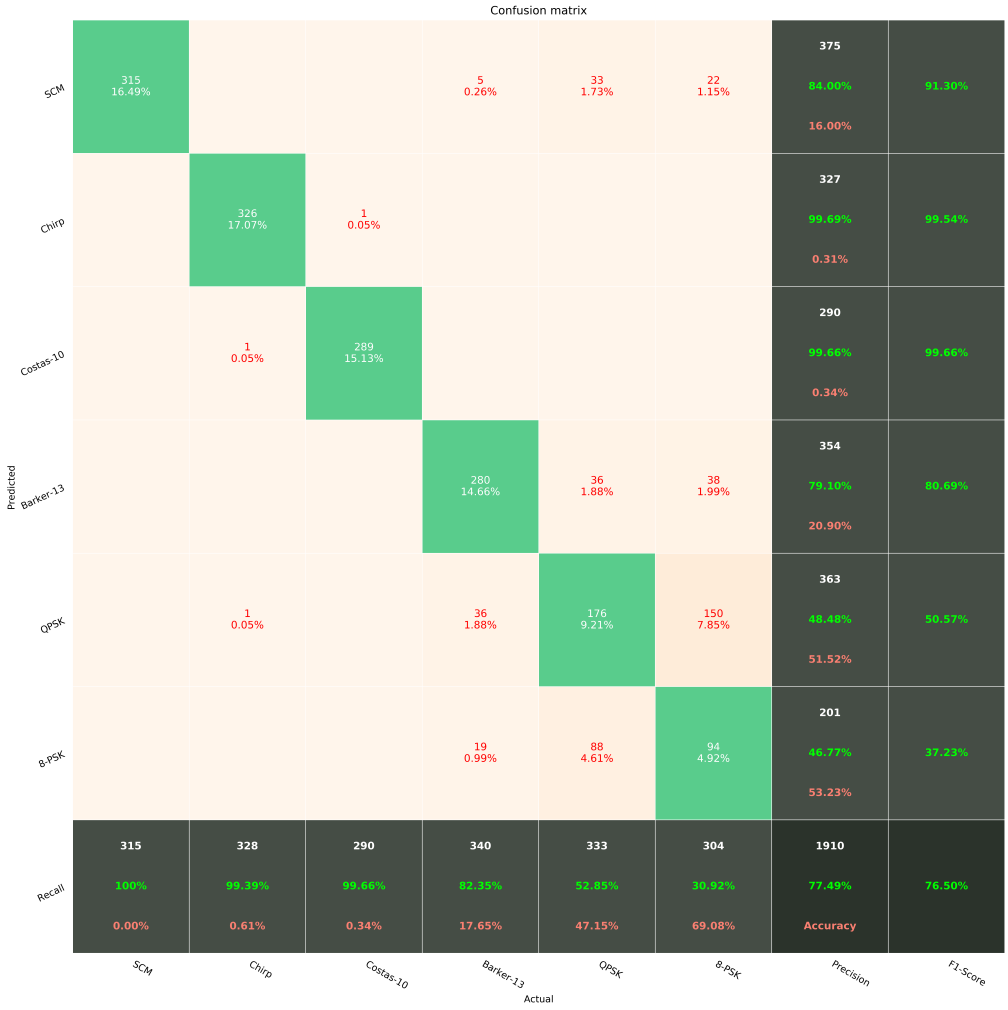
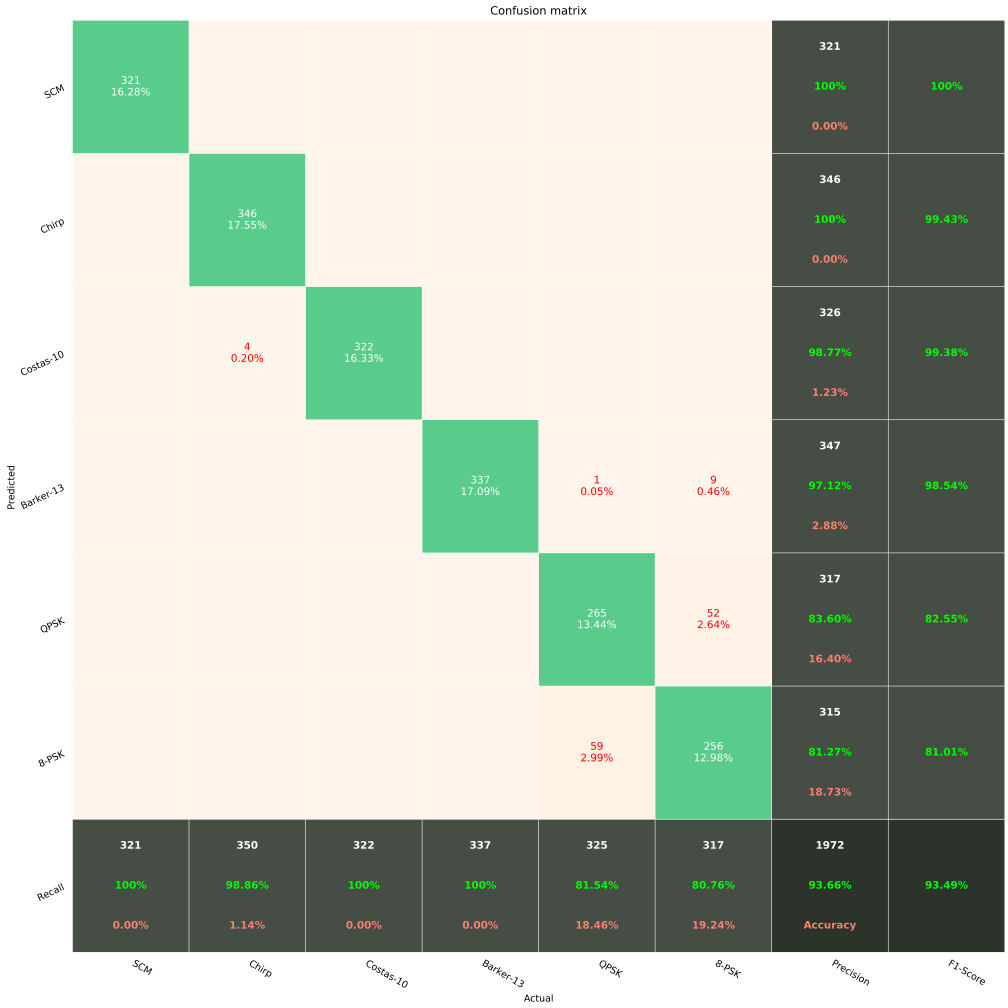


Figure 4.11: Confusion matrix of the CWTFI technique for test data of 6 class set at -10 dB SNR level.



(a)

Figure 4.12: Confusion matrix of the SincNet based model for test data of 6 class set at 10 dB SNR level.

(a)

Figure 4.13: Confusion matrix of the SincNet based model for test data of 6 class set at -10 dB SNR levels.

Confusion matrix plots at different SNR levels for IMOP classification results of the CNN, WVTFI, CWTFI, FFCNN and SincNet on 6 class set can be seen in Figures 4.6-4.13.

4.2.2 Pulse Detection Results

Figure 4.14 shows the signal detection result of the detection head of the proposed MODNET architecture under -20 dB SNR. It is observed that it can exhibit high detection performance even under intense noise.

In detection comparisons, window length of the Energy Detector is chosen as given in Section 5.2.1. Pulse detection comparison over varying SNR levels can be seen in Figure 4.15. As can be seen from the figure, MODNET is the top performing technique on all SNR levels. Energy Detector fails at detection below 0 dB SNR. Neural network based models perform better at low SNR levels.

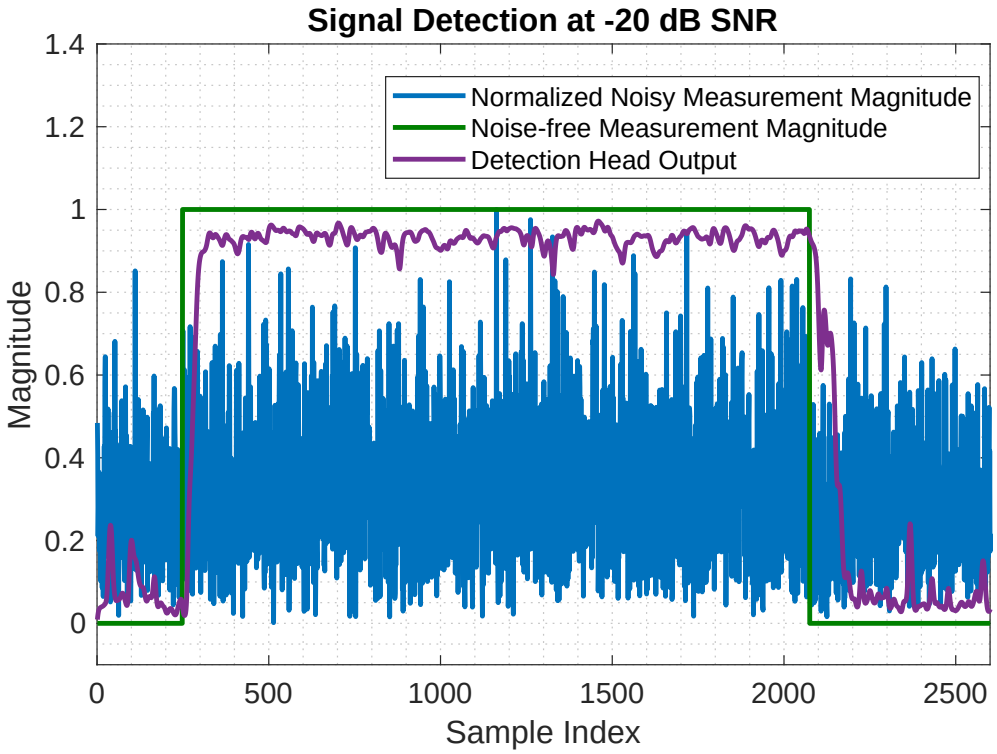


Figure 4.14: Pulse detection at -20db SNR with proposed MODNET structure.

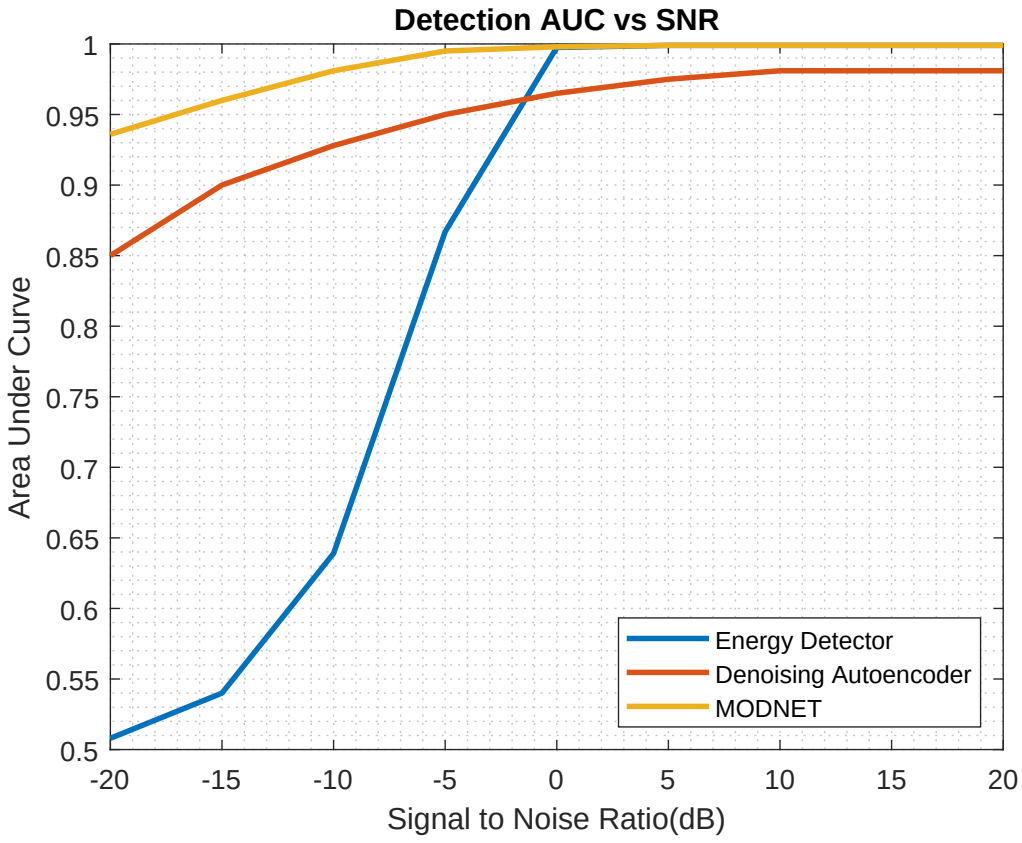


Figure 4.15: Area under ROC curve (AUC) values over varying SNR levels.

Chapter 5

Baselines

In this chapter, we discuss various solution approaches proposed for the pulse detection and IMOP classification tasks. We start with widely adopted neural network based classification methods, then we present traditional and neural networks based detection methods that are widely used in the literature.

5.1 Classification Techniques

In this section we first discuss the time-frequency based IMOP classification approaches, then give the details of a SincNet based waveform classification modality.

5.1.1 Time-Frequency Image based IMOP Recognition

Time-frequency representations(TFR) are more and more widely used for non-stationary signal analysis. They perform a mapping of a one-dimensional signal $x(t)$ into a two-dimensional function of time and frequency $TFR_x(t, w)$ in order to extract relevant information. Mostly used TFR methods are Short Time Fourier

Transform, Wigner-Ville Distribution, and Choi-William Distribution.

5.1.1.1 Short Time Fourier Transform

Let $F_x(t, w; z)$ denote the STFT of $x(t)$, given as:

$$F_x(t, w; z) = \int_{-\infty}^{\infty} x(s) z^*(s - t) e^{-jws} ds \tag{5.1}$$

where $z(t)$ is the windowing function controlling the desired time and frequency resolution of the resulting TFI.

5.1.1.2 Wigner-Ville Distribution

Let $WV_x(t, w)$ denote the Wigner-Ville(WV) distribution of $x(t)$, given as:

$$WV_x(t, w) = \int_{-\infty}^{\infty} x(t + s/2) x^*(t - s/2) e^{-jws} ds \tag{5.2}$$

WV distribution of a 1-D signal has good time-frequency resolution but is prone to cross-term interference.

5.1.1.3 Choi-William Distribution

Let $CW_x(t, w)$ denote the Choi-William(CV) distribution of $x(t)$, given as:

$$\begin{aligned} CW_x(t', w') = 2 \int \int_{-\infty}^{\infty} & \frac{\sqrt{\sigma}}{4\sqrt{\pi}|t|} e^{-w^2\sigma/(16t^2)} \\ & \times x(t' + w + \frac{t}{2}) x^*(t' + w - \frac{t}{2}) e^{-jw't} dt dw \end{aligned} \tag{5.3}$$

CW distribution acts as a low-pass filter which suppresses WVD cross-term interference.

5.1.1.4 Classification with Time Frequency Representations

In earlier methods [6], [7], [8], and [9], researchers have used traditional signal processing methods to extract features, such as Wavelet Analysis and instantaneous frequency analysis. However, since these methods are weak to characterize the distinctions among different types of radar signals, these methods are sensitive to noise and their generalization ability are limited. STFT, WVD and CW based Time-Frequency analyses present better distinction, that is, the frequency variation of radar waveforms with different modulation types. Some works [12], [2], [14] and [7] design a convolutional neural network(CNN) to recognize the modulations from the extracted TFIs .

5.1.2 SincNet based IMOP Recognition

Recently, [15] proposes a novel CNN architecture, called SincNet. Its overall architecture can be seen in Figure 5.1. In contrast to standard CNNs, that learn all elements of each filter, only low and high cutoff frequencies are directly learned from data with the proposed method. This offers a compact and efficient way to derive a customized filter bank specifically tuned for the desired application. Authors conducted experiments on both speaker identification and speaker verification tasks from raw samples and results were similar to CNN with less number of model parameters.

In this work we applied the same SincNet architecture to modulation type classification problem from raw measurements.

5.2 Detection Techniques

In this section we first define the problem of pulse detection, then present the details of energy based and autoencoder based pulse detector modalities.

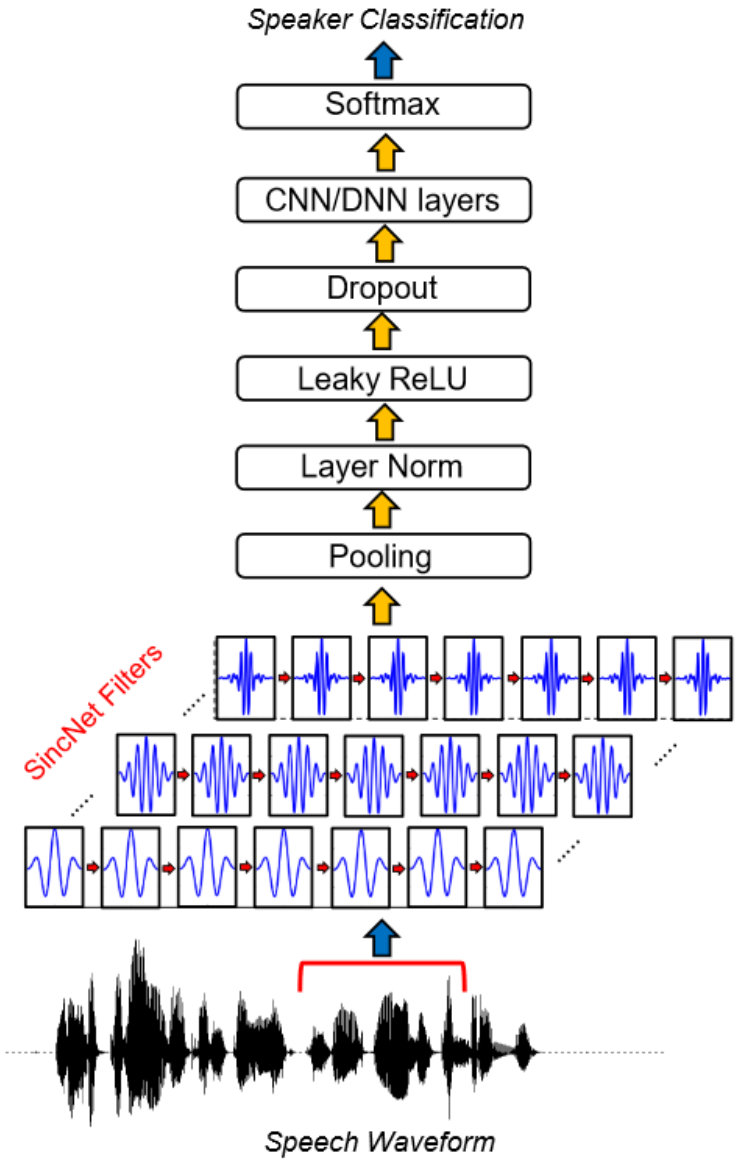


Figure 5.1: SincNet architecture. [15]

Two probabilities are of interest for the detection. Probability of detection, under hypothesis $\mathcal{H}_1$, defines the probability of correctly detecting present signal; and probability of false alarm, under hypothesis $\mathcal{H}_0$, defines the probability of falsely labeling the presence of signal. In the following sections, details of the baseline detection models are given.

5.2.1 Energy based Pulse Detector

The test statistic for energy detector is:

$$T(y) = \frac{1}{N} \sum_{n=1}^{N} |y(n)^2| \tag{5.4}$$

where N is window length, under hypothesis $\mathcal{H}_0$, $T(y)$ has chi-square distribution with $2N$ degrees of freedom for complex valued case and with N degrees of freedom for real valued case. For a large N, probability of false alarm under hypothesis $\mathcal{H}_0$ can be approximated by

$$P_f = Q(\frac{\epsilon - \sigma^2}{\sigma^2 \sqrt{N}}) \tag{5.5}$$

where $Q(.)$ is the complementary distribution function of the normal distribution, and σ^2 is the noise variance of the measurements. ϵ is the threshold that is related to probability of false alarm, given as:

$$\epsilon = Q^{-1}(P_f)\sigma^2\sqrt{N} + \sigma^2 \quad \text{(from [38])} \tag{5.6}$$

N must be carefully chosen in order not to miss signals with minimum pulse width that is required to be detected. If f_s is sampling frequency and minimum pulse width to be detected is PW_{min}, then N is the maximum integer not greater than $PW_{min} f_s$.

5.2.2 Autoencoder based Pulse Detector

Autoencoders are specific type of feedforward neural networks where the input is the same as the output. They compress the input into a lower-dimensional code and then reconstruct the output from this representation. The code is a compact "summary" or "compression" of the input, also called the latent-space representation.

An autoencoder consists of 3 components: encoder, code and decoder [30]. The encoder compresses the input and produces the code, the decoder then reconstructs the input only using this code. When the coding layer has a lower dimensionality than the input data, the autoencoder is said to be undercomplete. It will not be able to directly copy its inputs to the output, and will be forced to learn intelligent features to be able to form the output. This way, one can use an autoencoder to reconstruct the signal support from noisy measurements. After applying a simple threshold to the output of the autoencoder, signal support can be predicted.

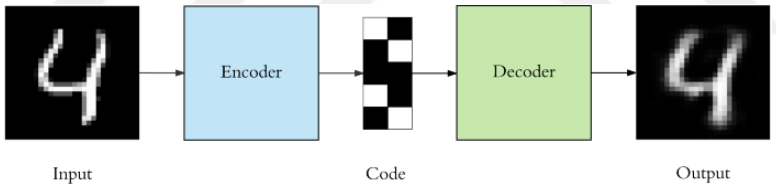


Figure 5.2: A reference architecture for denoising autoencoder. [39]

Chapter 6

Conclusion & Future Work

6.1 Conclusion

Simulation results show that the proposed FFCNN and MODNET techniques outperform the current state-of-the-art alternatives in all SNR levels and are easily scalable among broad range of modulation types. Moreover, promising results with real data are presented in [20].

For the pulse detection comparisons, proposed MODNET technique compared with energy detector and autoencoder based baseline. MODNET is the top performing technique on all SNR levels. Energy Detector fails at detection below 0 dB SNR. Neural network based models perform better at low SNR levels.

For the IMOP classification comparisons, proposed FFCNN (reassigned spectrum, robust least squares and CNN based) and MODNET (LSTM and multi-task learning based) techniques are compared with pure/time-frequency analysis based CNN models and a SincNet based model. FFCNN and MODNET are the top performing techniques on 0-20 db SNR region by a large margin. Below 0 db SNR, phase of the measurements start to get distorted largely. As a result,

FFCNN's phase related input becomes obsolete and FFCNN fails to differentiate phase modulations (reason of the sudden drop in accuracy). WV and CW based TFI representations are great for frequency related modulations but they are simply not enough to differentiate phase modulations (QPSK and 8-PSK). Moreover, WV TFI representation fails to extract meaningful features to differentiate frequency modulations below 0-5 dB SNR. Overall, MODNET performs better at all SNR levels.

6.2 Future Work

A possible research direction might be the additional post-processing blocks for the detection head of the proposed MODNET architecture to be able to get a more stable pulse support estimation on below 0 dB SNR levels. Moreover, convolutional LSTM based shared backbone architecture can be utilized.

Newer studies show promising results with GPT and transformer based generative neural network architectures. These type of backbones can be utilized for the proposed architectures for more comprehensive pulse detection and IMOP classification capabilities.

Bibliography

[1] D. Zhang, W. Ding, B. Zhang, C. Xie, H. Li, C. Liu, and J. Han, "Automatic modulation classification based on deep learning for unmanned aerial vehicles," *Sensors*, vol. 18, no. 3, p. 924, 2018.

[2] C. Wang, J. Wang, and X. Zhang, "Automatic radar waveform recognition based on time-frequency analysis and convolutional neural network," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pp. 2437–2441, IEEE, 2017.

[3] F. C. Akyon, Y. K. Alp, G. Gok, and O. Arikan, "Classification of intra-pulse modulation of radar signals by feature fusion based convolutional neural networks," in *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 2290–2294, IEEE, 2018.

[4] X. Feng, J. Yang, F. Luo, J. Chen, and X. Zhong, "Automatic modulation recognition by support vector machines using wavelet kernel," in *Journal of Physics: Conference Series*, vol. 48, p. 1264, IOP Publishing, 2006.

[5] C.-S. Park, J.-H. Choi, S.-P. Nah, W. Jang, and D. Y. Kim, "Automatic modulation recognition of digital signals using wavelet features and svm," *Advanced Communication Technology*, vol. 1, pp. 387–390, 2008.

[6] J. Lundén and V. Koivunen, "Automatic radar waveform recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 1, pp. 124–136, 2007.

[7] M. Zhang, L. Liu, and M. Diao, "Lpi radar waveform recognition based on time-frequency distribution," *Sensors*, vol. 16, no. 10, p. 1682, 2016.

[8] C. Wang, H. Gao, and X. Zhang, "Radar signal classification based on autocorrelation function and directed graphical model," in *Signal Processing, Communications and Computing (ICSPCC), 2016 IEEE International Conference on*, pp. 1–4, IEEE, 2016.

[9] T. Liu, Y. Guan, and Y. Lin, "Research on modulation recognition with ensemble learning," *EURASIP Journal on Wireless Communications and Networking*, vol. 2017, no. 1, p. 179, 2017.

[10] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *International conference on engineering applications of neural networks*, pp. 213–226, Springer, 2016.

[11] A. Ali, F. Yangyu, and S. Liu, "Automatic modulation classification of digital modulation signals with stacked autoencoders," *Digital Signal Processing*, vol. 71, pp. 108–116, 2017.

[12] S. Jeong, U. Lee, and S. C. Kim, "Spectrogram-based automatic modulation recognition using convolutional neural network," in *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 843–845, IEEE, 2018.

[13] Z. Zhou, G. Huang, H. Chen, and J. Gao, "Automatic radar waveform recognition based on deep convolutional denoising auto-encoders," *Circuits, Systems, and Signal Processing*, pp. 1–15, 2018.

[14] Z. Qu, X. Mao, and Z. Deng, "Radar signal intra-pulse modulation recognition based on convolutional neural network," *IEEE Access*, vol. 6, pp. 43874–43884, 2018.

[15] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with sincnet," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 1021–1028, IEEE, 2018.

[16] P. Q. Ly, S. Sirianunpiboon, and S. D. Elton, "Passive detection of bpsk radar signals with unknown parameters using multi-sensor arrays," in *Signal Processing and Communication Systems (ICSPCS), 2017 11th International Conference on*, pp. 1–5, IEEE, 2017.

[17] R. A. Dillard and G. M. Dillard, "Detectability of spread-spectrum signals," *ah*, 1989.

[18] Y.-C. Liang, Y. Zeng, E. C. Peh, and A. T. Hoang, "Sensing-throughput tradeoff for cognitive radio networks," *IEEE transactions on Wireless Communications*, vol. 7, no. 4, pp. 1326–1337, 2008.

[19] X. Yan, F. Long, J. Wang, N. Fu, W. Ou, and B. Liu, "Signal detection of mimo-ofdm system based on auto encoder and extreme learning machine," in *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1602–1606, IEEE, 2017.

[20] F. C. Akyon, Y. K. Alp, G. Gok, and O. Arikan, "Deep learning in electronic warfare systems: Automatic intra-pulse modulation recognition," in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, 2018.

[21] F. C. Akyon, Y. K. Alp, and G. Gok, "A method for classification of intra-pulse modulation of radar signals using feature fusion based convolutional neural network architecure," 2018. TR Patent, 2018-02050.

[22] F. C. Akyon, M. A. Nuhoglu, Y. K. Alp, and O. Arikan, "Multi-task learning based joint pulse detection and modulation classification," in *2019 27th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, 2019.

[23] F. C. Akyon, Y. K. Alp, and M. A. Nuhoglu, "A method for simultaneous signal detection, snr estimation and modulation classification using multi task learning based recurrent neural network architecture," 2019. TR Patent, 2019-05907.

[24] M. A. Nuhoglu, F. C. Akyon, and Y. K. Alp, "Yinemeli sinir ağı tabanlı otokodlayıcıyla radar sinyal tespiti," 2018. TR Patent, 2018/19484.

[25] P. E. Pace, *Detecting and classifying low probability of intercept radar*. Artech House, 2009.

[26] F. Auger and P. Flandrin, "Improving the readability of time-frequency and time-scale representations by the reassignment method," *IEEE Transactions on signal processing*, vol. 43, no. 5, pp. 1068–1089, 1995.

[27] J. A. Parker, R. V. Kenyon, and D. E. Troxel, "Comparison of interpolating methods for image resampling," *IEEE Transactions on medical imaging*, vol. 2, no. 1, pp. 31–39, 1983.

[28] C. Chen, "Paper 265-27 robust regression and outlier detection with the robustreg procedure," in *Proceedings of the Proceedings of the Twenty-Seventh Annual SAS Users Group International Conference*, 2002.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[30] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1. MIT press Cambridge, 2016.

[31] M. Wöllmer, M. Kaiser, F. Eyben, B. Schuller, and G. Rigoll, "Lstm-modeling of continuous emotions in an audiovisual affect recognition framework," *Image and Vision Computing*, vol. 31, no. 2, pp. 153–163, 2013.

[32] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[33] S. Zhang, Y. Wu, T. Che, Z. Lin, R. Memisevic, R. R. Salakhutdinov, and Y. Bengio, "Architectural complexity measures of recurrent neural networks," in *Advances in neural information processing systems*, pp. 1822–1830, 2016.

[34] J. P. C. Fonseca, "Fpga implementation of a lstm neural network," 2016.

[35] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[36] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[37] D. Kinga and J. B. Adam, "A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[38] M. Abramowitz and I. A. Stegun, "Handbook of mathematical functions with formulas, graphs, and mathematical table," in *US Department of Commerce*, National Bureau of Standards Applied Mathematics series 55, 1965.

[39] F. Chollet, "Building autoencoders in keras." https://blog.keras.io/building-autoencoders-in-keras.html. Accessed: 2020-07-25.