Instantaneous Frequency and Chirp Rate Estimation for Noisy Quadratic FM Signals by CNN

Huda Saleem Razzaq
Computer Science Department
Computer Science and Mathematics
Najaf, Iraq
hudas.alsarayefi@uokufa.edu.iq
orcid.org/0009-0003-5135-4837

Zahir M. Hussain
Computer Science Department
Computer Science and Mathematics
Najaf, Iraq
zahir.hussain@uokufa.edu.iq
orcid.org/0000-0002-1707-5485

DOI: http://dx.doi.org/10.31642/JoKMC/2018/100201

Received Oct. 20, 2022. Accepted for publication Nov. 12, 2022

***Abstract*—** *Deep learning and machine learning are widely employed in various domains. In this paper, Deep Neural Network (DNN) and Convolution Neural Network (CNN) are used to estimate the Instantons Frequency (IF), Linear Chirp Rate (LCR), and Quadratic Chirp Rate (QCR) for Quadratic Frequency Modulated (QFM) signals under Additive White Gaussian (AWG) noise and additive Symmetric alpha Stable (SαS) noise. SαS distributions are impulsive noise disturbances except for a few circumstances, lack a closed-form Probability Density Function (PDF), and an infinite second-order statistic. Geometric SNR (GSNR) is used to determine the impulsiveness of mixture noise for Gaussian and SαS noise. DNN is a machine learning classifier with few layers that reduce IF, LCR, and QCR estimation complexity and achieve high accuracy. CNN is a deep learning classifier that is built with multiple layers of IF, LCR, and QCR estimation. CNN is more accurate than DNN when dealing with large amounts of data and determining optimal features. The results reveal that SαS noise is substantially more damaging to IF, LCR, and QCR estimation than Gaussian noise, even when the magnitude is modest, and it is less damaging when alpha is greater than one. After training CNN for IF, LCR, and QCR estimation of QFM signals. The 2D-CNN model accuracy achieved 98.7603 and 1D-CNN is* 75.8678 *for ten epochs. DNN model accuracy achieved 37.5 for 1000 epochs. The accuracy of TFD (spectrogram & pspectrum) for frequency estimation of QFM signals was 38.4254 by spectrogram and 38.6746 by pspectrum.*

Keywords*— *Frequency estimation, QFM signal, sensors, Gaussian noise, SαS noise, TFD, CNN, DNN, machine learning, deep learning, ROC, and GSNR.

I. INTRODUCTION

Many practical signals utilized in RADAR, SONAR, medicinal applications, and wireless communications can be described as non-stationary Frequency Modulated (FM) signals [1], [2]. The Time-Frequency (TF) representation is widely acknowledged as the most suitable way of analyzing, characterizing, and processing such signals [3], [4]. Zhang Juan (2018) based on the time-frequency distribution and a convolutional neural network; this study developed a unique blind modulation classification approach CNN. This is the first attempt to approach the time-frequency map as a picture and recognize signals using an exceptional CNN-based technique from the computer vision field. To some extent, the combination provides a novel feature extraction approach that also corresponds to intuition. The simulation findings suggest that the method proposed in this work is efficient and robust and that it allows for a high degree of automation when extracting features, training weights, and making decisions [5]. Huda Saleem, and Zahir M. (2021) It is researched how to estimate the instantaneous frequency under additive white gaussian noise and additive symmetric α-stable noise. Based on the highest probability for the Short Time Fourier Transform, two TFD MATLAB algorithms pspectrum and spectrogram were investigated STFT. The results demonstrated that SαS noise is substantially more destructive for frequency estimation than Gaussian noise, even when the capacity is modest, and it is less destructive when alpha is more than one [6]. Safaa D., Zahir. M (2017) estimation of the instantaneous frequency of FM signals using TFD is of greater accuracy than methods using FT and correlation, which examines the performance of instantaneous frequency estimation of single-tone sinusoidal signals and IF estimation of FM signals in the presence of various types of colored noise. Two basic estimators are considered for single-tone sinusoidal signals: the maximum likelihood estimator employing the fast Fourier transform with interpolated peak estimation and the correlation approach [7]. Milczarek H. and A. Kawalec (2019) describe a novel approach for estimating the instantaneous frequency of an NLFM radar signal. There are

currently only a few limited reports in this field. The method employs a time-frequency plane combining short-time Fourier transform with IF curve smoothing. The technique estimates the instantaneous frequency of noise-buried signals to be within -3 dB. There is no substantial computing burden required due to the FFT and simple filtering application. The results of the simulation show that the technique is resilient and outperforms the well-known phase differentiation approach [8]. Zahraa C., and Zahir M. (2017) the performance of instantaneous frequency estimators of mono-component FM signals (a single-tone sinusoid) under AWGN with various types of multiplicative noise was investigated. Two basic estimators are considered for single-tone signals: maximum likelihood estimator employing discrete Fourier transform with interpolated peak estimate, and autocorrelation approach. The peak of a certain time-frequency distribution, the periodogram, has been addressed for linear and non-linear FM signals. Three statistical multiplicative noise models are investigated Gaussian, Rayleigh, and uniform. discrete Fourier transform is still better than the correlation for IF estimation, according to simulation results [9]. Akram J. (2020) investigates the instantaneous frequency estimation of multi-component signals in the time-frequency domain, using a combination of Eigen decomposition of time-frequency distributions and time-frequency filtering to extract signal components and estimate their instantaneous frequencies using the ridge detection and tracking procedure [10]. FM is a decreased antenna length that allows multiple transmissions within the same channel for different frequencies, as well as SNR reduction, which is essential in network communication systems. The rest of this paper is structured as follows: Section II is the problem statement. Section III introduces the aims and objectives. Section IV shows the methodology. Section V discusses a discussion of the results. Section VI is the conclusion.

II. PROBLEM STATEMENT

This work will discuss the estimation problem for FM signals under noise environments. Usually, Gaussian noise is considered. Impulsive noise is the real problem. An essential kind of impulse noise is the symmetric α-stable noise. Impulse noise is typically associated with Gaussian noise, making the estimation problem more difficult. We process to estimate the Instantaneous Frequency (IF) of nonlinear Frequency Modulated (FM) signals in the presence of a mixture of α-stable noise (a kind of non-Gaussian noise) and Gaussian noise.

III. AIMS AND OBJECTIVES

The aim is the instantaneous frequency and QCR of quadratic frequency modulated signals under noise environments, where the objectives or actions you will take to achieve this aim are: First, the frequency-modulated sinusoidal waves generate such as QFM signals. Second, these signals are affected by a mixture of noise AWGN and SαSN. Third, TFD includes spectrogram and pspectrum applied to the analytical noisy signals, where noisy signals are analyzed by HT. fourth, DNN is applied to noisy signals. Finally, CNN includes 2D-CNN models applied for the noisy input signals.

IV. METHODOLOGY

This work is the first proposal to solve the problem of estimating the instantaneous frequency by using a convolutional neural network under the noise environment, where SαS noise is impulsive noise making the estimation problem more difficult. As for DNN and TFD, they were used for comparison.

Noisy QFM signals are analyzed. We used the STFT for IF estimation to compare the classical method (STFT) and deep learning method (CNN). Wigner-Ville Distribution (WVD) is better distributed for Pure QFM, but under noise, its performance degrades due to the cross-term effect, which may disturb the maximum (peak) of the IF law. So, STFT is more suitable for the IF estimation of noisy QFM signals. Deep Neural Networks (DNN) and Convolutional Neural Networks (CNN) are a class of Artificial Neural Networks (ANN). CNN is a deep learning classifier, designed with many layers, and proved to be more accurate than DNN when dealing with noisy data and finding optimal features.

The time-frequency distribution method and the convolutional network and deep neural network method are used to solve the problem of estimating the instantaneous frequency of noisy FM signals, where the processing tools: are TFD, DNN, CNN, AWGN, and SαSN. The work context includes the following: First, generating the frequency-modulated signals as described in section 1. Second, generating the Gaussian noise and the SαS noise and combining them to generate the noisy FM signals as described in sections 2, 3, and 4, respectively. Third, the estimation of the instantaneous frequency of the noisy FM signals by TFD includes two methods spectrogram and pspectrum as described in section 5. Fourth, the estimation of the instantaneous frequency of noisy FM signals by DNN and CNN is shown in sections 6 and 7 respectively. Applications of this work are RADAR and medical SONAR. The effects are better localization for RADAR and better diagnosis in medical SONAR. The improvement of this work is the accurate and fast estimation of IF. The achievement of work was achieved by utilizing a personal laptop using MATLAB 2022b, where Central Processing Unit (CPU) Intel(R) Core (TM) i5, 7th and Random Access Memory (RAM) 8.00 GB.

1. INSTANTANEOUS FREQUENCY AND FREQUENCY MODULATION

The instantaneous frequency, which characterizes the changes in frequency content across time is a necessary aspect of FM transmissions. A signal's IF is a time derivative of its instantaneous phase (θ(t)) as follows [11-13]:

$$f_i(t) = \frac{1}{2\pi}\frac{d\theta(t)}{dt} \tag{1}$$

$$\theta = 2\pi(f_o t + \delta\frac{t^2}{2} + \rho\frac{t^3}{3}) \tag{2}$$

Quadratic Frequency Modulation (QFM) signal has also been considered in this work with quadratic IF law as follows:

$$s(t) = A\, e^{j2\pi\left(f_o t+\frac{\delta}{2}t^2+\frac{\rho}{3}t^3\right)} \tag{3}$$

Where δ is the linear modulation index, f_o is the initial frequency (in Hertz), and A being the amplitude, ρ is the quadratic modulation index of the QFM signal, with the quadratic IF law:

$$f_i(t) = f_o + \delta t + \rho t^2 \tag{4}$$

2. ADDITIVE WHITE GAUSSIAN NOISE

The probability density function (PDF) of additive white Gaussian noise is as follows, with zero mean and variance (power) σ^2:

$$p(n) = \frac{1}{\sigma\sqrt{2\pi}} e^{-n^2/2\sigma^2} \quad (5)$$

where n is a random variable and σ is the standard deviation of the noise. This noise is generated in MATLAB by wgn function as follows:

$$n = wgn(M, N, pndB) \quad (6)$$

where N is the length of signal, M realizations of noise, $pndB$ is $SNR\ in\ dB$.

3. SYMMETRIC α-STABLE NOISE

Symmetric α-Stable distribution noise necessitates the use of four parameters (α, γ, β, and μ), with the characteristic function defined as [14,15]:

$$\psi(\omega) = \exp(-\gamma|\omega|^{\alpha}) \quad (7)$$

where $(0 < \alpha \leq 2)$ is often referred to as the tail index or characteristic exponent. When $\alpha < 2$ is present, the distribution is algebraic-tailed with a constant tail α, implying infinite variance. As the density of tails increases smaller, they become heavier. The SαS distribution is simplified to the Gaussian distribution when $\alpha = 2$. The SαS distribution is simplified to the Cauchy distribution when α = 1 and β = 0. The SαS distribution is simplified to the lévy distribution when α = 0.5 and β = 1. The parameter γ>0, often known as the dispersion, is a positive constant associated with the distribution scale. The parameter serves a similar function to the variance in a second-order process. The skewness parameter is β ∈ [−1, 1]. μ ∈ R is the location parameter. We generated symmetric α-Stable (na) in MATLAB as follows:

$$na = random('Stable'.a.0.g.0.[M\ N]) \quad (8)$$

where N is the length of the signal, M realizations of noise, a is alpha, g is the scale parameter.

4. MIXTURE AWG NOISE WITH SαS NOISE

A non-stationary signal is one with a varying frequency content over time. Non-linear FM signals affected by AWGN and $S\alpha SN$. $S\alpha S$ noise is geometric power, while Gaussian noise is fixed power. Geometric SNR (GSNR) is used to calculate noise impulsiveness, which is defined by zero-order statistics. The conventional SNR does not apply since all 2nd order moments are limitless. The scale parameter of $S\alpha SN$ is defined as follows:

$$\gamma = \sqrt{p_S / C^{\left(\frac{2}{\alpha}-1\right)}} \quad (9)$$

Where C is the exponential of Euler's constant, $C = e^{Ec} \approx 1.7811$, $p_S = \frac{p_T}{1+b}$, b is the ratio between Gaussian and SαS noise, p_T is total noise power, and $(0 < \alpha \leq 2)$ is alpha or characteristic exponent.

5. IF ESTIMATION BY TFD

TFD is a double transform from the time domain to the time-frequency domain that describes the Fourier transform of an analytical signal's instantaneous autocorrelation. The simplest formula for a time-frequency distribution is the Short-Time Fourier Transform (STFT), which is a windowed frequency distribution [16, 17]. TFD IF estimates the presence of noise (AWGN and SαSN) in FM transmissions. Find the STFT spectrogram (spec (t, f)). Then, using the peak (max) of the spec, calculate the IF as follows:

$$\hat{f}_i(t) = \arg(\max\{spec(t, f)\}); 0 \leq f \leq \frac{f_s}{2} \quad (10)$$

For each GSNR, compute the relative squared error as follows:

$$e = \left|\left(\hat{f}_i \times df - \mathrm{IF}_t\right)/f_o\right|^2 \quad (11)$$

Where f_o fundamental frequency, $\hat{f}_i$ estimated frequency, IF_t theoretical IF with spectrogram timing, $df = \frac{f_s}{N}$ and $N = 1024$. TFD estimated IF using spectrogram and pspectrum MATLAB functions; pspectrum differs from spectrogram in segment lengths, overlapping segments, and windows. Spectrogram length equals $1 \times \left[\frac{N}{2} + 1\right]$. Pspectrum length equal $1 \times N$. Pspectrum controls the length of the segments and the overlap between consecutive segments using the time resolution and overlap percent pair parameters; it divides the signal into overlapping segments and applies a Kaiser window to each segment.

6. IF ESTIMATION BY DNN

DNN is used to estimate Instantons IF, LCR, and QCR. DNN is included in the first structure and components are the input layer, the number of hidden layers, an output layer, the number of nodes in each layer, weights, and nodes. Second forward and backward stages. Third, the parameters are bias, learning rate, and the number of iterations. For IF classification, we can use a neural network. An activation function and an optimization procedure are required in DNN. The activation function is a set of mathematical operations performed on the output. The activation functions are chosen based on the sort of problem that the network will be solving. The most popular activation functions are the Sigmoid or logistic function and the Hyperbolic tangent or tanh function [18, 19]. The Adam approach is based on integrating the benefits of two methods, AdaGrad and RMSProp. The advantages of Adam include invariant magnitudes of parameter updates, suitability for problems with big data sets, and suitability for problems with noise [20]. The network was used by the following requirements:

- *Structure of DNN*

The DNN is the content input layer, two hidden layers, output layer. The number of nodes in the input layer is twenty. The number of nodes in the first & second hidden layers is three. Design DNN to frequency, LCR, and QCR classification.

- *Forward Stage*

In this stage applied many steps. First, find the sum of the product for the input node and weight, then summation with bias. Second, applied activation function for the sum of products, where the activation function of the first & second hidden layer is ReLU. The activation function of the output layer is softmax. Third, compute cost function is cross-entropy. The error is the most important term in the loss expression. It shows neural network fits the dataset. There are many cost functions including mean squared error, normalized squared error, weighted squared error, and cross-entropy error. This work used cross entropy error,

where to compute error for each node in the output layer and compute loss value. A neural network is stopped training when the error, or the difference between the desired and expected output, is less than a certain threshold value, or when the number of iterations or epochs exceeds a certain threshold value.

- *Backward stage*

In this stage need an optimization algorithm to update the parameters (weights and bias), and need compute the error ratio for each layer. The optimization algorithm controls how the parameters of the neural network are adjusted. This work used Scale Conjugate Gradient (SCG) optimization algorithm. Where the learning rate is 1e-4, it is used in the backward stage for learning improvement. Epoch equals 1000. SCG makes use of second-order neural network information while requiring just O(N) memory use, where N is the number of weights in the network [21].

- *Generated QFM signals for DNN training*

Noisy QFM signals are generated with a GSNR range [-50: 10: 50]. The total number of input signals is 308, with each signal's length 149, where the number of frequencies, LCR, QCR equal 10, and realizations equal 7. The dataset was randomly divided into a training ratio of 85%, a validation ratio of 5%, and a test ratio of 10%. Then, dataset training by DNN forward and backward stages. After, getting to optimal parameters, we will compute DNN performance for the samples test by some measures such as ROC and accuracy. The frequency range is computed as follows:

- Initial frequency is $f1 = 10$, last frequency is $f2 = 19$
- The number of frequencies is $f = 4$, different frequency is $df = \left\lfloor \frac{f2-f1}{nf} \right\rfloor$
- Range IF is $fr = [\, f1 : increasing\ by\ df : f2\,]$

The LCR range is computed as follows:

- The initial slope is $e1 = 0.1$, the final slope is $e2 = 0.9$
- The number of slopes is $ne = 4$, different frequency is $de = \frac{e2-e1}{ne}$
- Range LCR is $er = [e1 : increasing\ by\ de : e2\,]$

The QCR range is computed as follows:

- The initial QCR is $q1 = -0.9, and$ The final QCR is $q2 = -0.1$
- The number of slopes is $nq = 4$, Different QCR is $dq = \frac{q2-q1}{nq}$
- Range QCR is $qr = [q1 : increasing\ by\ dq : q2\,]$

7. IF AND CR ESTIMATION BY CNN

In CNN, it is included the first architecture and components. The second two stages of training are the forward and backward stages. Third, are the parameters for the training network. The architecture and components are included the input layer, the number of layers in the feature extraction stage, the number of layers in the classification stage, weights (filters), and nodes. The parameters two types are key parameters and hyperparameters. Key parameters are bias, learning rate, and weights. Hyperparameters are the number of filters, mini-batch, and epochs. Deep learning CNN models are composed of several layers, including a convolution layer with filters, a pooling layer, a Fully Connected (FC) layer, and the Softmax function. Describe the work of each layer in a convolutional neural network [22, 23]. The network was used by the following requirements:

- *Architecture CNN*

The CNN is a content input layer that includes images with a size of 80×80. The hidden layer includes convolution, batch normalization, ReLU, max pooling, fully connected, and dropout. The output layer includes a fully-connected, softmax, classification layer. The number of nodes in the input layer is 6400. The number of nodes in the hidden layer is based on the parameter for each layer. The number of the node of the output layer equals ten, it represented the number of classes. Design CNN for frequency & slope classification. Convolution layer with the number of filters equal 30, size filter is 3×3, the stride is $[1\ 1]$, and padding same. Other convolution layers were different in the number of filters, where it equals $60, 90, and\ 128$. Max Pooling with size 2×2, stride $[2\ 2]$, padding $[0\ 0\ 0\ 0]$. Fully Connected with 100 output nodes in the feature extraction stage. The dropout layer ratio is 50%. Fully Connected equals ten output nodes in the classification stage. The classification layer used cross-entropy.

- *Forward Stage*

In this stage, many steps are applied as the following: First, input images, where input size is 80×80×1, output size 80×80×1, and hyperparameter is normalization zero-center.

Second, find the feature map by computing the sum of the product for the input node with weighted (filters), then summation with bias. In this layer input size is 80×80×1, the output size is 80×80×30, and hyperparameters are $FS = 3 \times 3$, $NF = 30. S = 1. P = 1. C = 1$, and the total of parameters is 300, where FS is filter size, NF is the number of filters, S is stride, P is padding, C is the number of channels for input sample, a total of parameters is No. weights & No. bias.

Third, batch normalization is applied, it speeds up training by halving or more epochs. Fourth, the activation function is applied for the sum of products, where the activation function is ReLU.

$$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \quad (12)$$

$$y_i = \gamma \hat{x}_i + \beta \quad (13)$$

where input size of this layer is 80×80×30, output size 80×80×30, hyperparameters are offset γ =1, Epsilon ε =0, scale β =0, variance σ^2, and mean μ.

Fourth, max pooling was applied, it was used to decrease the size of the feature maps. As a result, the amount of calculation performed in the network is reduced, where the input size is 80×80×30, and the output size is 80×80×30.

Fifth, the convolution layer is applied three times with the number of filters $(60, 90, 128)$, then each of them follows with the ReLU layer and max pooling layer. When the number of filters is 60 the input size is 40×40×30, the output size is 40×40×60, the hyperparameter is $NF = 60$, $S = 1, P = 1$, C=30, $FS = 3 \times 3$, and a total of parameters is 16260. The max pooling layer input size is 40×40×60, the output size is 20×20×60. When the number of filters is 90 the input size is 20×20×60, the output size is 20×20×90, the hyperparameter is $NF = 90$, $S = 1.\ P = 1$, C=60, $FS = 3 \times 3$, and a total of

parameters is 48690. The max pooling layer input size is 20×20×90, the output size is 10×10×90. When the number of filters is 128 the input size is 10×10×90, the output size is 10×10×128, the hyperparameter is $NF = 128,\ S = 1, P = 1$, C=90, $FS = 3 \times 3$, and a total of parameters is 103808. The max pooling layer input size is 10×10×128, the output size is 5×5×128.

Sixth, fully connected applied, it is fed forward neural networks, where all of the inputs from the previous layer are linked to each node in the next layer. The reason for the two-layer fully connected used is to perform better and it increases the capacity of the network. The input size of this layer is 5×5×128=3200, the output size is 100, the hyperparameter is No. of output nodes, and the total of parameters is 320100.

Seventh, the dropout layer is applied, it is a method of stochastic regularization. It aids in the prevention of overfitting and the accuracy and loss will gradually improve. Dropout is often placed after connected layers because they have the most parameters.

Finally, the classification stage includes three layers that are fully connected, softmax, and classification output. The fully connected ten output nodes represent the number of classes, it also feeds forward neural networks, where the input size is 100, the output size is 10, the hyperparameter is the number of classes, and the total of parameters is 1010. The softmax is utilized as the activation function in the neural network output layer for multi-class classification tasks. It has a value between zero and one; and provides interpretable probabilities for the categories. The classification output applied; it's included Categorical Cross Entropy (CCE) is a loss function that is used in multi-class classification tasks. Training loss is good when loss (cross-entropy) equals zero. It is used one-hot encoded for true labels. The validation loss of CNN is computed by categorical cross-entropy.

$$\mathrm{CCE} = -\frac{1}{N}\sum_{i}^{N}\sum_{j}^{M} y_{ij}\log\left(p_{ij}\right) \qquad (14)$$

Where N is the number of rows, and M is the number of classes. To get on the output network after completing training, used two methods last-iteration and best-validation-loss. The last iteration returns the network after the last training iteration. The best validation loss is to return the network after the training iteration with the lowest validation loss. To use this option, you must specify the validation data in the training option. In this case, compare validation loss with validation data. If validation loss is greater than the slope threshold (validation data in option CNN) or access to No. of iterations, go to the backward stage, else stop training and return the output network. The measurements are used to select the best models include accuracy, precision, recall, F-measure, and Receiver Operating Characteristic curves (ROC) a useful tool for assessing the performance of a machine learning or deep learning model [24].

- *Backward Stage*

  In this stage need an optimization algorithm to update the parameters, and need compute the error ratio for each layer. The optimization algorithm controls how the parameters of the neural network are adjusted. This work used Adaptive moment estimation (Adam).

- *Parameters of CNN*

  Deep learning algorithms rely heavily on parameters such as bias, learning rate, weights, and other parameters. An optimization approach is frequently used to calculate model parameters. Key parameters are automatically estimated from the data and the model. Hyperparameters are manually set and used in procedures to aid in the estimation of model parameters. Learning rate equal 1e-4, it is used in the backward stage to learning improvement. Max epochs = 100, mini-batch size = 8, and Adam used initially learn rate= 0.001, Epsilon= 0.00000001, squared gradient decay factor = 0.999, and gradient decay factor = 0.9000. Hyperparameters in convolution are the number of filters, size of the filter, and padding. The batch normalization is included ε, β, and γ. The max pooling is included in size and stride. The dropout layer is included probability. The fully connected includes several classes.

- *Generated signals of CNN*

  Noisy QFM signals are generated with a Geometric SNR range $[-50\ \ 50]$. In this work, the dataset is generated with non-linear FM signals under additive white Gaussian noise and symmetric stable noise, also IF, LCR, and QCR are used as explained in section eighth but used No. of IF, LCR, and QCR equal ten. CNN deals with 2-Dimensional images, so after generating noisy signals, they are converted into 2-Dimensional images. The total number of input signals is 12120, with each signal's length 28×28. There are ten classes including signals with IF, LCR, and QCR, each class has 1212 samples. The dataset was randomly divided into a training ratio of 95%, a validation ratio of 5%, and a test ratio of 10%. Then, dataset training by CNN forward and backward stages. After optimal parameters are gotten, we will compute CNN performance for test samples by some measures such as accuracy, precision, recall, and F1_score.

V. DISCUSSION OF THE RESULTS

The network learns from the input data and predicts the IF and CR. TFD, DNN, and CNN models were used to simulate IF, LCR, and QCR estimation for noisy QFM signals. As for the metrics for measuring the efficiency of the training model or the performance efficiencies of the prediction algorithms, they are accuracy, Precision, recall, F-score confusion matrix, and ROC. The results show high accuracy for parameter estimation by confusion matrix and some measures such as accuracy, precision, and F1-score. SαS is an impulsive model, where alpha is more harmful even if it is of small value, where it affects the frequency, LCR, and QFM guesses. A variable b determines the ratio of AWGN to SαS. Fig. (1) shows the test error rate for FE of noise QFM by spectrogram. Fig. (2) shows the test error rate for FE of noise QFM by spectrum. Fig. (3) shows relative absolute error for IF, LCR, and QCR estimation of noisy QFM signals by DNN. Fig. (4) shows ROC for Noise QFM by DNN. Fig. (5) shows a confusion matrix for QFM signals by DNN. The confusion matrix rows correspond to the predicted class and the columns correspond to the target class. The diagonal cells correspond to observations that are correctly classified. The off-diagonal cells correspond to incorrectly classified observations. The column on the far right of the plot shows the percentages of all the samples predicted to belong to each class that is correctly and incorrectly classified. The row at the bottom of the plot shows the percentages of all the samples belonging to each class that is correctly and incorrectly classified. The cell in the bottom right of the plot shows the overall accuracy. Fig. (6) shows ROC for noise QFM by 2D-CNN. The ROC curves are an important tool for evaluating the performance of a machine-learning model. The ROC curve

shows the relationship between the True Positive Rate (TPR) for the model and the False Positive Rate (FPR). The best classifications will show the receiver operating line hugging the left and top sides of the plot axis. Fig. (7) shows a confusion matrix for QFM signals by 2D-CNN. Fig. (8) shows relative absolute error for IF, LCR, and QCR estimation of noisy QFM signals by 2D-CNN. Fig. (9) shows a relative absolute error of IF estimation for noise QFM by 2D-CNN and TFD Spectrogram (TFDs) & Pspectrum (TFDp). Fig. (10) shows epochs Vs. entropy for noisy QFM signals by ANN. Fig. (11) shows the accuracy and loss rate for noisy QFM by 2D-CNN. Fig. (12) shows FE, LCR, & QFM of the noisy QFM signals by using 1D-CNN. Fig. (13) shows the ROC of the noisy QFM signals by using 1D-CNN. Fig. (14) shows the ROC of the noisy QFM signals by using 1D-CNN. Fig. (15) shows the confusion matrix for QFM signals by 1D-CNN. Tab. (1) shows measures of the 2D-CNN & ANN models for noisy QFM signals. Deep CNN outperformed artificial neural networks and TFD in estimating the instantaneous parameters of non-stationary signals. In time-frequency distribution, spectrogram and pspectrum were utilized, with the findings indicating that pspectrum is superior to spectrogram for IF estimation.

VI. CONCLUSION

This research provided an overall description of the performance of machine learning, TFD, and deep-learning algorithms (CNN) for estimating the frequency, LCR, and QCR for noisy Non-Linear Frequency-Modulated signals (NLFM). Under additive white Gaussian noise and symmetric stable noise. It examines the IF, LCR, and QCR estimate inaccuracy under various GSNRs. In DNN, just two hidden layers are employed. The CNN model includes many layers, and the simple structure built for the DNN or CNN model serves to reduce the communication system's complexity, power consumption, and cost. The simulation results demonstrate that alpha is more detrimental than beta, even if it has a tiny disability, and it has a substantial effect on guess frequency, linear chirp rate, and quadratic chirp rate.

The future works are generating the noisy signals by asymmetric α-stable noise or using other types of signals and applying the proposed system to them, such as heart and brain signals. The proposed system limitation is required powerful hardware, high processing for the processor, and storage memory.

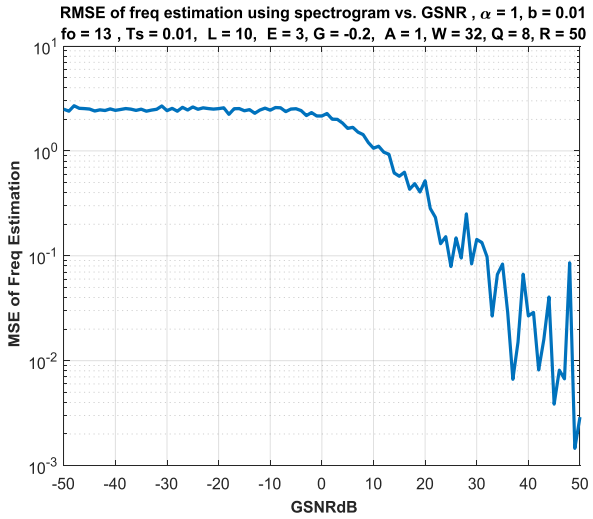

Fig. 1. Test Error Rate for FE of Noise QFM by (Spectrogram)

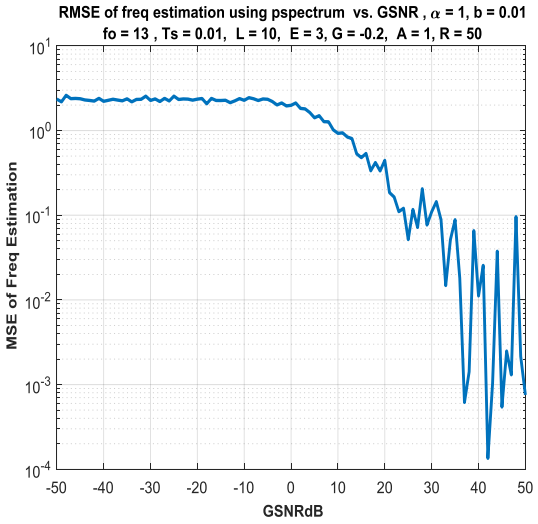

Fig. 2. Test Error Rate for FE of Noise QFM by (Pspectrum).

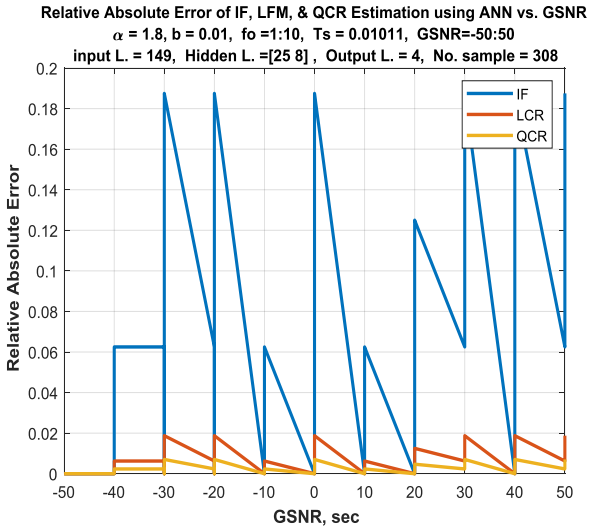

Fig. 3. Relative Absolute Error for IF, LCR, QCR Estimation of Noisy QFM Signals by DNN.

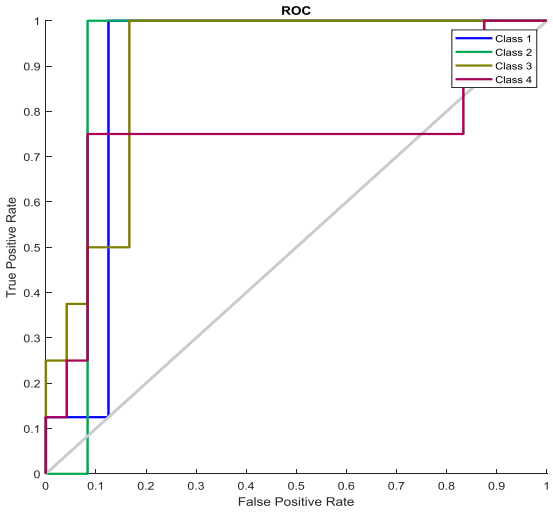

Fig. 4. ROC for Noise QFM by DNN.

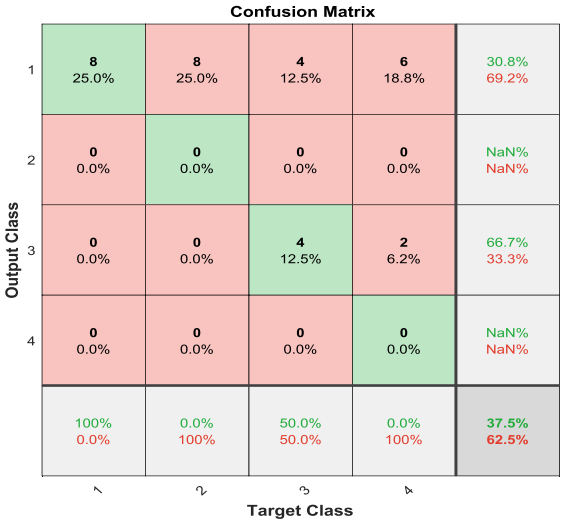

Fig. 5. A Confusion Matrix for QFM Signals by DNN.

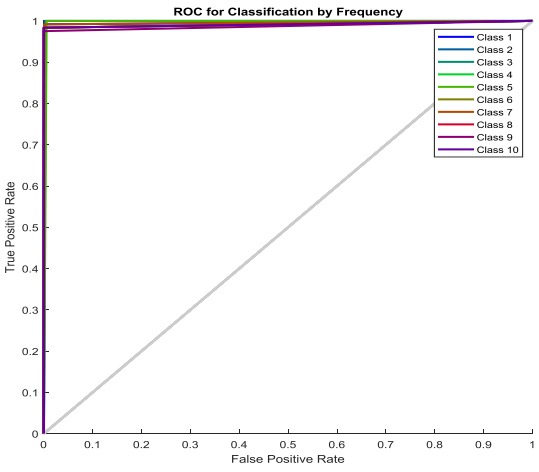

Fig. 6. ROC for Noise QFM Signals by 2D-CNN.

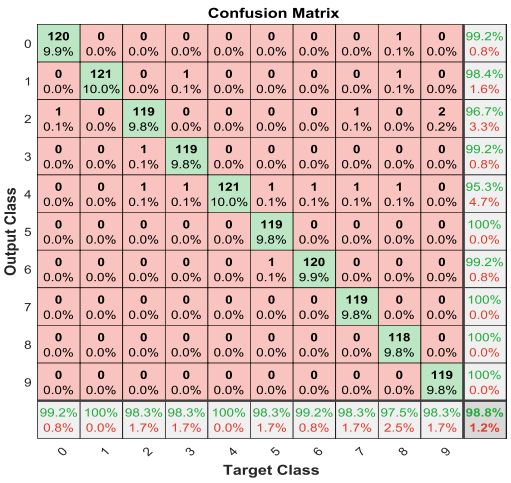

Fig. 7. A Confusion Matrix for QFM Signals by 2D-CNN.

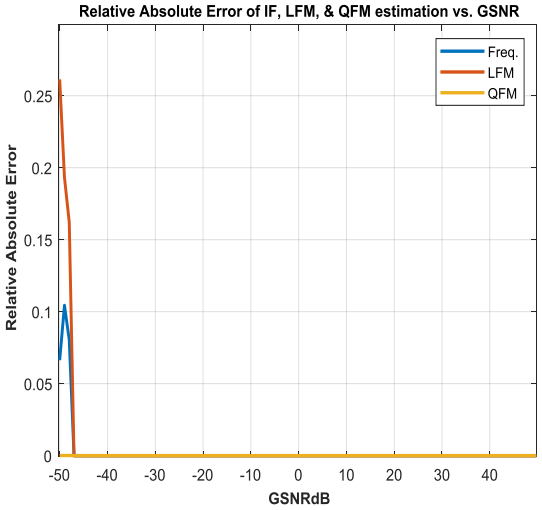

Fig. 8. Relative Absolute Error for IF, LCR, & QCR Estimation of Noisy QFM Signals by 2D-CNN.

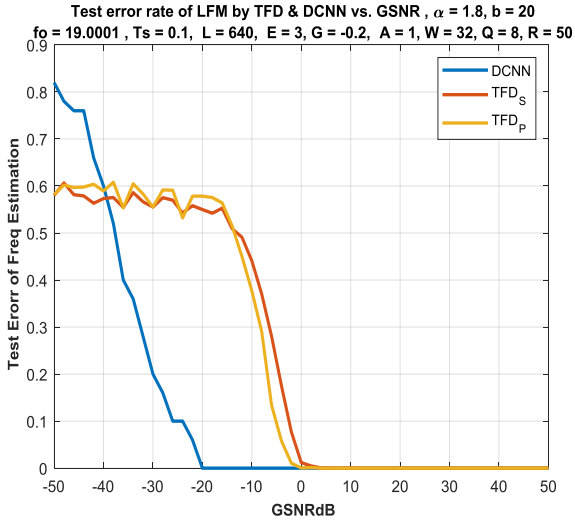

Fig. 9. Relative Absolute Error of IF Estimation for Noise QFM by 2D-CNN and TFD Spectrogram (TFDs) & Pspectrum (TFDp).

TABLE I. MEASURES OF CNN & ANN MODEL FOR NOISY QFM SIGNALS.

Measures	2D-CNN	1D-CNN	DNN
Accuracy	98.7603	75.8678	37.5000
Precision	98.7911	73.2224	50
Recall	98.7603	75.8678	0
F1_Score	98.7757	74.5216	0
Epoch	10	10	1000
Time (Sec)	5629	3872	4.73

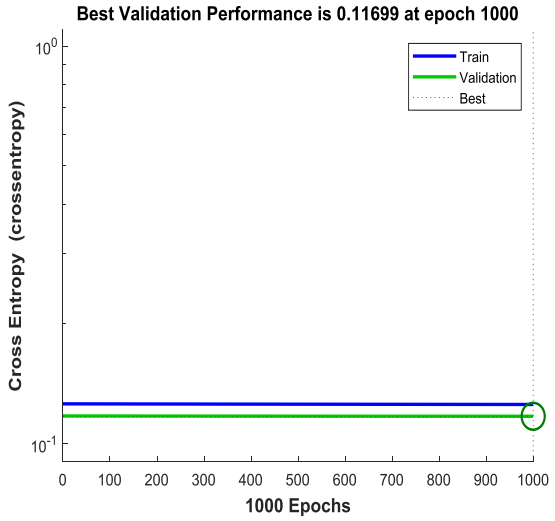

Fig. 10. Epochs Vs. Entropy for Noisy QFM Signals by ANN.

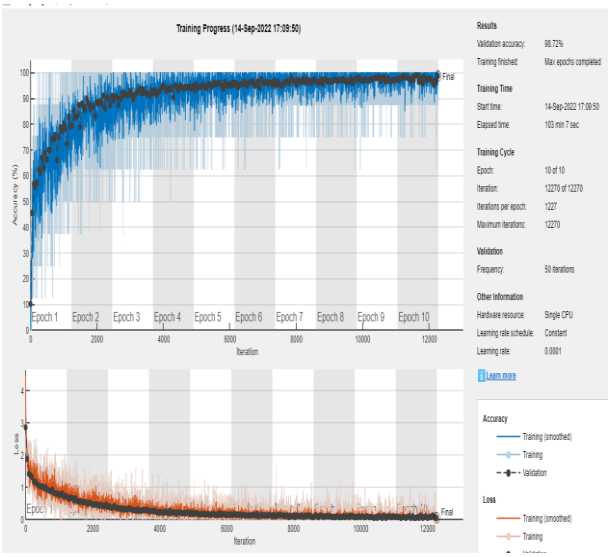

Fig. 11. Accuracy and Loss Rate for Noisy QFM by 2D-CNN.

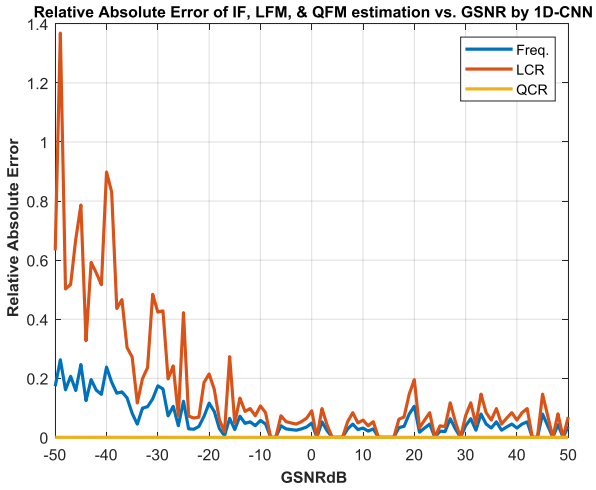

Fig. 12. FE, LCR, & QFM of the Noisy QFM Signals by Using 1D-CNN.

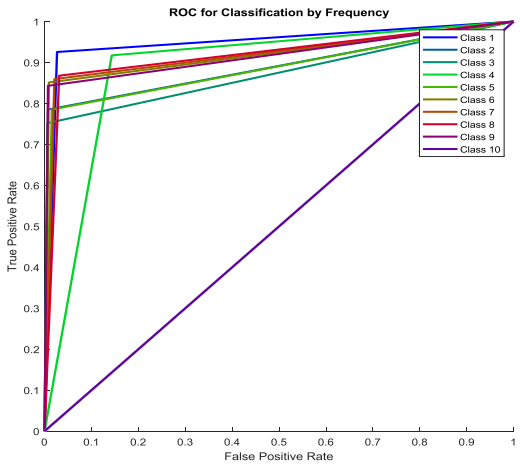

Fig. 13. ROC of the Noisy QFM Signals by Using 1D-CNN.

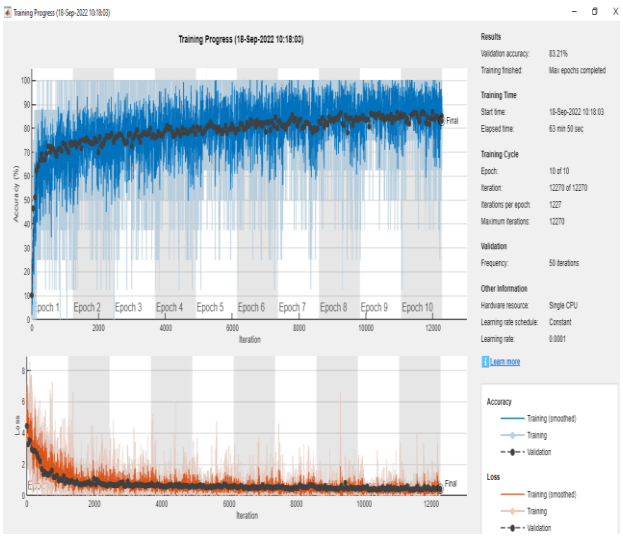

Fig. 14. ROC of the Noisy QFM Signals by Using 1D-CNN.

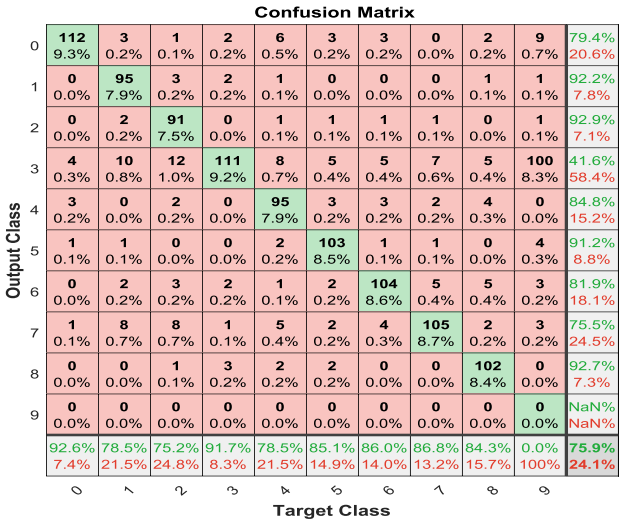

Fig. 15. A Confusion Matrix for QFM Signals by 1D-CNN.

REFERENCES

[1] B. Boashash, "Estimating and interpreting the instantaneous frequency of a signal part 1: Fundamentals," Proc. IEEE, vol.80, no. 4, pp. 520–538, Apr. 1992.

[2] A. Papandreou-Suppappola (Ed.), Applications in TimeFrequency Signal Processing, CRC Press, 2002.

[3] L. Cohen, "Time-frequency distributions – A review," Proc. IEEE, vol. 77, no. 7, pp. 941–981, July 1989.

[4] B. Boashash, "Estimating and interpreting the instantaneous frequency of a signal – Part 2: Algorithms and applications," Proc. IEEE, vol. 80, no. 4, pp. 540–568, Apr. 1992.

[5] Zhang Juan, Yong Li, and Junping Yin, "Modulation classification method for frequency modulation signals based on the time-frequency distribution and CNN." IET Radar, Sonar & Navigation 12.2 (2018).

[6] Razzaq Huda Saleem, and Zahir M. Hussain. "Instantaneous Frequency Estimation for Frequency-Modulated Signals under Gaussian and Symmetric α-Stable Noise." 2021 31st International Telecommunication Networks and Applications Conference (ITNAC). IEEE, 2021.

[7] Al-Khafaji Safaa D., Zahir M. Hussain, and Katrina Neville. "Frequency estimation of FM signals under non-Gaussian and colored noise." International Journal of Applied Engineering Research 12.22 (2017).

[8] Milczarek H., and A. Kawalec. "Instantaneous frequency estimation for radar NLFM signal using combined STFT and TFD ridge smoothing technique." XII Conference on Reconnaissance and Electronic Warfare Systems. Vol. 11055. SPIE, 2019.

[9] Al-Rammahi, Zahraa C., and Zahir M. Hussain. "Instantaneous Frequency Estimators of FM Signals: Performance under Models of Multiplicative Noise." Global Journal of Pure and Applied Mathematics 12.6 (2016).

[10] Khan, Nabeel Ali, et al. "Novel direction of arrival estimation using adaptive directional spatial time-frequency distribution." Signal Processing 168 (2020): 107342.

[11] Liu Xuelian, and Chunyang Wang. "A novel parameter estimation of chirp signal in α-stable noise," IEICE Electronics Express, 2017.

[12] Milczarek, Hubert, et al. "Estimating the Instantaneous Frequency of Linear and Nonlinear Frequency Modulated Radar Signals—A Comparative Study," Sensors 21.8, 2021.

[13] B. Boashash, P. O'Shea, M. J. Arnold, "Algorithms for instantaneous frequency estimation: a comparative study," Proc. SPIE, 1990.

[14] Liu M., Han Y., Chen Y., Song H., Yang Z., & Gong, F., "Modulation Parameter Estimation of LFM Interference for Direct Sequence Spread Spectrum Communication System in Alpha-Stable Noise," IEEE Systems Journal, 2020.

[15] Zhang G., Wang J., Yang G., Shao Q., & Li, S., "Non-linear processing for correlation detection in symmetric alpha-stable noise," IEEE Signal Processing Letters, 25(1), 120-124, 2017.

[16] Almayyali H. R., & Hussain Z. M., " Deep Learning versus Spectral Techniques for Frequency Estimation of Single Tones: Reduced Complexity for Software-Defined Radio and IoT Sensor Communications". Sensors, 21(8), 2729, 2021.

[17] B. Boashash, Ed., "Time-Frequency Signal Analysis and Processing: a comprehensive Reference", Elsevier, Oxford, UK, 2016.

[18] Bengio Y.," Learning deep architectures for AI". Foundations and trends® in Machine Learning, 2(1), 1-127, 2009.

[19] Alpaydin E., "Introduction to machine learning". MIT Press, 2009.

[20] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," pp. 1–15, 2014.

[21] Mishra, Soumya, Rajashree Prusty, and Pradosh Kumar Hota. "Analysis of Levenberg-Marquardt and Scaled Conjugate gradient training algorithms for artificial neural network-based LS and MMSE estimated channel equalizers." 2015 International Conference on Man and Machine Interfacing (MAMI). IEEE, 2015.

[22] LeCun Y., Bengio Y., Hinton G., "Deep learning", Nature, 521, 436, 2015.

[23] Amato G., Carrara F., Falchi F., Gennaro C., Meghini C., Vairo C., "Deep learning for decentralized parking lot occupancy detection", Expert Syst, 72, 327–334, 2017.

[24] Powers D. M., "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation",2011.