 doi:10.20944/preprints202208.0068.v1

Article

Instantaneous Frequency Estimation of FM Signals under Gaussian and Symmetric α-Stable Noise: Deep Learning versus Time-Frequency Analysis

Huda Saleem Razzaq [1], Zahir M. Hussain [2,*]

[1] Computer Science and Mathematics, University of Kufa, Najaf 54001, Iraq; hudasaleem.hs93@gmail.com
[2] School of Engineering, Edith Cowan University, Joondalup, WA 6027 Australia
* Correspondence: z.hussain@ecu.edu.au or zmhussain@ieee.org

Abstract: Deep Learning (DL) and Machine Learning (ML) are widely used in many fields, but rarely used in Frequency Estimation (FE) and Slope Estimation (SE) of signals. Frequency and slope estimation for Frequency-Modulated (FM) and single-tone sinusoidal signals are essential in various applications, such as wireless communications, sonar, and radar measurements. In this work, artificial neural network (ANN) and convolutional neural network (CNN) are used in frequency and slope estimation for FM signals under Additive White Gaussian Noise (AWGN) and Additive Symmetric alpha Stable Noise (SαSN). SαS distributions are impulsive noise disturbances found in many communication environments like marine systems; their distribution lacks a closed-form Probability Density Function (PDF), except for specific cases, and infinite second-order statistic, hence Geometric SNR (GSNR) is used in this work to determine the impulsiveness of noise in a mixture of Gaussian and SαS noise processes. ANN is a machine learning classifier, designed with few layers for reducing FE and SE complexity while getting higher accuracy as compared with classical techniques. CNN is a deep learning classifier, designed with many layers for FE and SE, and proved to be more accurate than ANN when dealing with big data and finding optimal features. Simulation results show that SαS noise can be much more harmful for FE and SE of FM signals than Gaussian noise. DL and ML can significantly reduce FE complexity, memory cost, and power consumption, which is important in many systems such as some Internet of Things (IoT) sensor applications. After training DCNN for frequency and slope estimation of LFM signals, the performance of DCNN (in terms of accuracy) can give acceptable results at very low signal-to-noise ratios where TFD fails, giving more than 20dB difference in the GSNR working range.

Keywords: Frequency estimation; FM; sensors; Internet of Things (IoT); software-defined radio (SDR); alpha-stable noise; time-frequency distribution; deep learning

1. Introduction

Frequency estimation is utilized in various engineering applications, including communications, RADAR, frequency identification of sinusoidal signals, and resonance sensing systems [1, 2]. Many signals in practice are nonstationary, such as Frequency Modulation (FM) that is signal found in communication and other application. Those signals can be classified as either mono-component or multicomponent signals. Machine learning and deep learning methods are important in many fields such as frequency classification, intrusion detection system (IDS) [3], landslide detection [4], Software Defined Networking (SDN) [5], smart logistics [6], and a gait type classification [7], that allow communication between devise and run task; which led to Internet of Things (IoT). When devices are increased the amount of data increases (big data). Machine learning is used to analyze these data and to deal with efficiently to make meaningful and valid decisions [8]. Liu Jinyu (2020) proposes an algorithm for frequency estimation of sinusoidal FM signal using an interpolation of Fast Fourier Transform (FFT) and Discrete-Time Fourier Transform

(DTFT), relying on N-point FFT to find the position of the maximum FFT; then three spectral lines located within the main lobe are used to estimate the frequency [9]. Akram J. (2020) studies the instantaneous frequency estimation of multi-component signals within the time-frequency domain, where a combination of Eigen decomposition of time-frequency distributions and time-frequency filtering is used to extract signal components and estimate their instantaneous frequencies using the ridge detection and tracking procedure [10]. Xu S., & Shimodaira H. (2019) offer a method that relies on neural networks to F0 estimate. It includes two sub tasks as a classification to determine whether or not the frame has voice, and a regression for estimating the (F0) value. Single model is used for both; output is (F0) values for voice frames and zero for unvoiced frames [11]. Bruno Silva et al. (2018) proposes to estimate the Doppler frequency by using Artificial Neural Network (ANN). The results explain that this method has a better performance and lower computational cost than the traditional methods like Robust Chinese Remainder Theorem (RCRT). They use an ANN with 3 neurons in input layer (remainders by RCRT), 10 neurons in hidden layer, and 17 neurons in output layer. It is randomly divided into 3 parts: 60% is used for training, 10% is used for validation and 30% is used for test [12]. Chen Xiaolong, & et al. (2019) introduce a model that relies on convolutional neural network to signal frequency signal and LFM signal detection and estimation. The pre-trained model is based on signals with 2-dimensional domain and content multiple convolutional layers, pooled layers and fully connected layers, and finally softmax classification is used as the output layer [13]. In various applications, such as wireless communications and image processing, the Symmetric α-Stable (SαS) noise is widely encountered. Liu Xuelian et al. (2017), analyze the characteristics of α-stable noise, and the chirp signal in α-stable noise is converted into Gaussian-like distribution. Then, using fractional Fourier transform to estimate the initial frequency and chirp rate of signal in α-stable noise [14]. Generally, it is difficult to estimate the parameters of FM signal under a mixture of α-stable noise (which is a type of non-Gaussian noise) and Gaussian noise.

The rest of this paper is outlined as follows: Section 2 introduces problem. Section 3 introduces objective. Section 4 introduces instantaneous frequency and FM. Section 5 introduces additive white Gaussian noise. Section 6 presents symmetric α-stable noise. Section 7 introduces machine learning. Section 8 introduces deep learning. Section 9 introduces metrics. Section 10 and 11 introduces IF estimation based on DNN & CNN, and TFD. Section 12 discusses the results, and Section 13 presents the conclusion of the paper.

1. Problem

Two fundamental issues in signal processing are signal estimation and signal separation of nonstationary signals. In signal estimation issue, we estimate the Instantaneous Frequency (IF) of Frequency Modulated (FM) signals under α−stable noise. FM signals are used in many engineering applications, such as in radar, sonar, and communications. Such signals contain the intended information in the frequency content. In telecommunications when information bearing signal is sent through a communication channel, transmitted FM signal is corrupted with noise or interference (other signals in the same communication channel). At the receiving end, it needs to recover the intended information from receive signal. In case of FM signal, intended information is the frequency content. Accurate frequency estimation leads to accurate recovery of the true information [39, 10].

2. Objectives

The problem of Frequency Estimation (FE) is processed by classical techniques such as Fourier and correlative techniques. Moreover, the same problem is processed by deep neural networks and CNN. This work aims to provide an accurate and fast estimation of IF and instantons slope, thus deep learning for frequency classification is promising. The proposed method has RADAR and medical sonar applications, where radar functions

 doi:10.20944/preprints202208.0068.v1

include range (localization), angle, and velocity. Medical sonar functions include diagnosis, classification, and tracking. The consequences are improved radar localization; improved medical sonar diagnosis. Frequency Modulate (FM) is reduced antenna length, allowing multiple transmission within the same channel for different frequencies, and SNR reduction which is so important in network communication system.

3. Instantaneous Frequency and FM

The instantaneous frequency, which describes the frequency content's variations with time, is an essential characteristic of FM signals. The IF of a signal is a derivative of its instantaneous phase $(\theta(t))$ concerning time is [15, 16]:

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

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

In this work, the signal model having Linear Frequency Modulation (LFM) law is [17]:

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

where α is the linear modulation index, f_o is the initial frequency (in Hertz), and A being the amplitude. Using Eq. (1), the LFM signal IF will be [18]:

$$f_i(t) = f_o + \alpha t \tag{4}$$

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

$$s(t) = A\, e^{j2\pi\left(f_o t + \frac{\alpha}{2}t^2 + \frac{\beta}{3}t^3\right)} \tag{5}$$

where β is the quadratic modulation index of the QFM signal, with the quadratic IF law:

$$f_i(t) = f_o + \alpha t + \beta t^2 \tag{6}$$

4. Additive White Gaussian Noise

Additive White Gaussian (AWG) noise has the following probability density function (PDF) with zero mean and variance (power) σ^2 [19]:

$$p(n) = \frac{1}{\sigma\sqrt{2\pi}}e^{-n^2/2\sigma^2} \tag{7}$$

where n is a random variable and σ is the standard deviation of the noise.

The procedures of generating AWGN is as follows:

1. Calculating the power contained in the input signal (x), were

$$p_x = \frac{1}{L}\sum_{i=0}^{L-1} |x[i]|^2, \quad L = \text{length}(x) \tag{8}$$

2. Converting the supplied $SNRdB$ (SNR in dB) to a linear scale and finding the noise power in terms of SNR and signal power (p_x), were

$$SNR = 10^{SNRdB/10}; \quad N_0 = p_x/SNR \tag{9}$$

3. Using the following equations to determine the AWG noise:

$$G_n = \sigma \times n.\ \text{if } X \text{ is real} \tag{10 a}$$

$$G_n = \sigma \times (n + i\, m).\ \text{if x is complex} \tag{10 b}$$

where $n, m \in \mathcal{N}(0, \sigma^2)$. For a real signal $\sigma = \sqrt{N_0}$, for a complex signal $\sigma = \sqrt{N_0/2}$.

5. Symmetric Alpha-Stable Noise

Symmetric α-Stable distribution noise requires 4 parameters (α, γ, β, and μ), with characteristic function defined as [20, 21]:

$$\psi(\omega) = \exp(-\gamma|\omega|^{\alpha}) \tag{11}$$

where ($0 < \alpha \leq 2$) is also known as the tail index or characteristic exponent. When $\alpha < 2$, the distribution is algebraic-tailed with a constant tail α, meaning infinite variance. The density tails become heavier as it gets smaller. When $\alpha = 2$, the SαS distribution is reduced to the Gaussian distribution. When $\alpha = 1$ and $\beta = 0$, the SαS distribution is reduced to the Cauchy distribution. When $\alpha = 0.5$ and $\beta = 1$, SαS distribution is reduced to the lévy distribution. The parameter $\gamma>0$, usually called the dispersion, is a positive constant related to the distribution scale. The parameter γ plays a role that is analogous to that of the variance for a second-order process. Skewness parameter is $\beta \in [-1, 1]$. Location parameter is $\mu \in R$. The procedures of SαS are as follows:

1. For β=0 any symmetric alpha-stable noise (N_S), then generating a random variable (V) uniformly distributed, and independent exponential random variable (W) as follows:

$$V = \frac{\pi}{2} \times (2u - 1) \tag{12}$$

$$W = -\log(v)) \tag{13}$$

where $u, v \in \mathcal{U}$, the standard uniform distribution.

$$N_S = \frac{\sin(\alpha \times V)}{\{\cos(V)\}^{1/\alpha}} \times \left[\frac{\cos(V \times (1-\alpha))}{W}\right]^{(1-\alpha)/\alpha} \tag{14}$$

2. For $\alpha \neq 1$, generating a random variable (V) uniformly distributed, and independent exponential random variable (W) as follows:

$$V = \pi \times (u - 0.5) \tag{15}$$

$$W = -\log(v)) \tag{16}$$

$$N_S = S_{\alpha,\beta} \times \frac{\sin\{\alpha(V + B_{\alpha,\beta})\}}{\{\cos(V)\}^{1/\alpha}} \times \left[\frac{\cos\{V - \alpha(V + B_{\alpha,\beta})\}}{W}\right]^{(1-\alpha)/\alpha} \tag{17}$$

$$S_{\alpha,\beta} = \left\{1 + \beta^2 \tan^2\left(\frac{\pi\alpha}{2}\right)\right\}^{1/(2\alpha)} \tag{18}$$

$$B_{\alpha,\beta} = \frac{\arctan\left(\beta \tan\frac{\pi\alpha}{2}\right)}{\alpha} \tag{19}$$

When scale and shift are applied as equation, we have

$$N_{SS} = \sigma N_S + \mu \tag{20}$$

3. For $\alpha = 1$, generating a random variable (V) and (W) as above:

$$N_S = \frac{2}{\pi}\left\{\left(\frac{\pi}{2} + \beta V\right)\tan V - \beta \log\left(\frac{\frac{\pi}{2} W \cos V}{\frac{\pi}{2} + \beta V}\right)\right\} \tag{21}$$

When scale and shift are applied as equation, we have:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 August 2022 doi:10.20944/preprints202208.0068.v1

$$N_{ss} = \sigma N_S + \frac{2}{\pi}\beta\sigma\log\sigma + \mu \tag{22}$$

6. Machine Learning

ML is kind of artificial intelligence technique which can automatically detect beneficial information from huge datasets. Machine Learning systems can be classified according to the amount and type of supervision they get during training. There are many classification algorithms [22] like Support Vector Machines (SVMs) [23], Naïve Bayesian (NB) [24], K-Nearest Neighbors (KNN) [25], Decision Tree (DT) [26], and Artificial Neural Network (ANN). This work is based on artificial neural network to frequency and slope classification. ML is divided into four types: supervised learning, unsupervised learning, Semi-supervised, and reinforcement learning, this work is based on supervised learning.

1. A supervised learning algorithm takes a known set of features and known responses to the data (decision) and trains a model to generate reasonable predictions for new data. It is mainly used in classification algorithms and regression algorithms. The aim of the supervised learning is to construct a model that makes predictions based on evidence in the presence of uncertainty [27].
2. Unsupervised learning It is used to draw inferences from a dataset consisting of features without a decision. When new data is introduced, it uses the previously learned features to recognize the decision of the data. It is mainly used for clustering techniques and feature reduction [28].
3. Semi-supervised is the type of ML used that combines supervised and unsupervised by combination unlabeled data and small amount labeled data.
4. Reinforcement learning is a type of learning which makes decisions based on which procedures give a more positive result. The learner has no knowledge which procedures to take until it has been given a situation.

Artificial Neural Network (ANN) is a kind of ML, it imitates the way human brains work. It contains an input layer, many hidden layers, and an output layer. The nodes in neighboring layers are fully connected. ANN includes a large number of nodes; it has strong ability for recognizing nonlinear functions. ANN with complex structure has training is time-consuming. ANN types are single layer neural networks and multi-layer neural networks. Single layer is neurons connect from input layer to output layer; it cannot include hidden layer. In multi-layer networks, single hidden layer is called shallow neural network, while two or more layers are called deep neural network [29]. This work is based on deep neural network.

In ANN, there is a need for activation function and optimization algorithm. Activation function is mathematical operations run on the output. The activation functions are chosen depending on the type of problem to be solved by the network. The most common of activation functions are Sigmoid or logistic, and Hyperbolic tangent or tanh [30].

Optimization algorithm is used to calculates the weights update, its four types are Gradient Descent (GD), RMSProp, Adam, and Levenberg-Marquardt optimization. GD is very popular optimization technique in machine learning. There are three types of GD: Batch, Mini-batch, and Stochastic Gradient Descent (SGD). Levenberg-Marquardt in ANN training and Adam in CNN training are used in this study. SGD is finding the error for each training data and adjusting the weights [31].

$$\Delta W_{ij} = \eta\ \delta_i x_j \tag{23}$$

where η is Learning rate, δ_i is generalized delta rule, and output from the input node j, $\delta_i = e_i$ if linear activation function, $\delta_i = \varphi'(v_i)e_i$ if non-linear activation function, φ' is derivative of activation function and v_i weighted sum of output node i.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 August 2022 doi:10.20944/preprints202208.0068.v1

Adaptive Moment Estimation (Adam) is the method used for computing adaptive learning rates for each parameter. Adam method is design depends on combining the advantages of two methods AdaGrad and RMSProp. Advantages of Adam are magnitudes of parameter updates are invariant, proper for problems that have large data, appropriate for problems with noise. The procedures of Adam are as follows [32].

1. Inputs: $\alpha.\beta_1.\beta_2.f(\theta).\theta_0$, where $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$, $\mathcal{G}_t$ is gradients and $\beta_1^t.\beta_2^t$ we denote β_1 and β_2 of power t.
2. Output: θ_t.
3. Initial parameters: $m_0 = 0.v_0 = 0.t = 0$.
4. while θ_t not converged do {

$t = t + 1;$ $\quad\quad \mathcal{G}_t = \nabla_\theta f_t(\theta_{t-1})$

$m_t = \beta_1.m_{t-1} + (1 - \beta_1).\mathcal{G}_t;$ $\quad\quad v_t = \beta_2.v_{t-1} + (1 - \beta_1).\mathcal{G}_t^2$

$\tilde{m_t} = m_t + (1 - \beta_1^t);$ $\quad\quad \tilde{v_t} = v_t + (1 - \beta_2^t)$

$\theta_t = \theta_{t-1} - \alpha.\tilde{m_t} \big/ (\sqrt{\tilde{v_t}} + \epsilon)$

7. Deep Learning

Deep learning is subset of machine learning. Deep learning techniques are used for big-data process such as image pattern recognition, speech recognition and synthesis, etc. It is required for CPU power increase, and powerful GPUs. The word "deep" refers to the large number of hidden layers that include the neural network [33]. Deep learning relies on Convolutional Neural Networks (CNN). A CNN is a kind of ANN that consists input layer, output layer, and hidden layers [34]. Deep learning models consist of various deep networks, such as deep neural networks (DNNs), deep brief networks (DBNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs) which are supervised learning models, but generative adversarial networks (GANs), auto encoders, and restricted Boltzmann machines (RBMs) which are unsupervised learning models. DNN works with multiple hidden layers and it works on 2-Dimensional (2D) data, thus the input data must be transformed into 2D matrices for frequency or slope detection. Convolutional Neural Networks (CNNs), the name "convolutional neural network" indicates that the network employs a mathematical operation called convolution. Convolution is a specialized kind of linear operation. Convolution help improve a machine learning system. There are two types of convolutions: valid and same. Deep learning CNN models train and test, each input image will pass it through a series of convolution layers with filters (Kernels), Pooling, Fully Connected layers (FC) and apply Softmax function to classify an object [35, 31]. CNN converts the manual methods for extracting features into automatic processes. DCNN architecture is illustrated in Figure (1).

1. Convolutional layer: It computes feature map as follows:

$$\mathrm{OFea}(x.y.f) = \mathrm{AF}\left(\sum_{v=0}^{c}\sum_{i=0}^{k}\sum_{j=0}^{k} A(x+i.y+j.v) \times W(i.j.v.f) + b(f)\right) \quad (24)$$

where $i.j$ index of filter, v number of channels, f number of filters, AF is activation function.

Three hyperparameters control the size of the output volume: the depth is number of filters, stride (S slide the filter) and zero-padding (P zeros around the border). Calculating number of neurons as follow:

$$H = W = \frac{n - f + 2 \times P}{S} + 1 \quad (25)$$

where f is the size of filter and n is the size of input image.

$$p = \frac{f-1}{2} \quad (26)$$

2. ReLU: is Rectified Linear Unit. The ReLU activation function performs depending on zero threshold.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 August 2022 doi:10.20944/preprints202208.0068.v1

$$f(x) = \max(0.x). \tag{27}$$

3. Pooling layers: It reduces the number of parameters. Pooling can be of different types: Max, mini, and average pooling. Max pooling takes the maximum element from feature map, Mini take the minimum element from feature map, and average take the average element from feature map.

$$D^{(L)} = D^{(L-1)} \tag{28}$$

$$H^{(L)} = \frac{H^{(L-1)} - F^{(L)}}{S^{(L)}} + 1 \tag{29}$$

$$W^{(L)} = \frac{W^{(L-1)} - F^{(L)}}{S^{(L)}} + 1 \tag{30}$$

where D is depth of filters; H & W is high and width of images, L is layer, $L-1$ is previous layer, $F^{(L)}$ size of filters, and $S^{(L)}$ stride.

4. Fully Connected Layer: It is ANN, its input is 1D-array, where Flattening is converted data into vector, all neurons in layer have full connections to all nodes in the previous layer.
5. Softmax layer: It is used probabilities associated with many classes, where probabilities summation equal one. Computing softmax layer is as follow:

$$\text{Softmax}\,(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \tag{31}$$

6. Classification layer: It takes input value from the Softmax layer and assigns into one class by using the cross-entropy function. Cross entropy measure of different between actual outputs and predict outputs of the training data. It used as a loss function.

$$E(\hat{y}.y) = -\sum_{i=1}^{C} \hat{y}_i \log(y_i) \tag{32}$$

Where C number of classes, $\hat{y}_i$ is actual value and y_i predicted value.

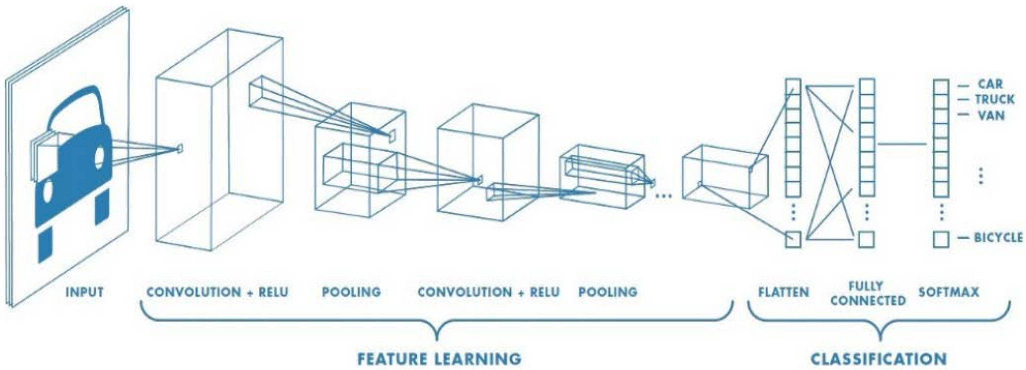

Figure 1. An illustration of the CNN architecture.

Reduction of Overfitting in DNN: Overfitting usually happens when the amount of the used parameters (the capability of the network) is much larger than the number of training samples. A model with the problem of overfitting makes great predictions for training samples but poor ones for validation data. There are two ways to reduce the overfitting which are dropout and Data augmentation [36]. Dropout is introduced by Srivastava et al [37], which means that each hidden neuron with probability 0.5 settings to zero.

Data augmentation: It is well-known that DNNs need to be trained on a large number of training samples to achieve satisfactory prediction and prevent overfitting. Data augmentation is a simple and commonly-used method to artificially enlarge the dataset by methods such as: random crops, intensity variations, horizontal flipping, etc. [41]. Training DNN parameters are learning rates that improve learning output. Backpropagation is used to update weights, it relies on optimal algorithms, and epoch is training iterative for all training data.

8. Metrics

Many metrics are applied to evaluate machine learning and deep learning methods. The perfect models are chosen by using these measures [38].

1) Accuracy (Ac) is the rate of correctly classified samples to overall samples.

$$\mathrm{Ac} = \frac{TP+TN}{TP+FP+FN+TN} \tag{33}$$

2) Precision (P) is rate of true positive samples to predicted positive samples.

$$P = \frac{TP}{TP+FP} \tag{34}$$

3) Recall (R) is rate of true positive samples to total positive samples.

$$R = \frac{TP}{TP+FN} \tag{35}$$

4) F-measure (F) is average of the precision and the recall.

$$F = \frac{2*P*R}{P+R} \tag{36}$$

5) False Negative Rate (FNR) is rate of false negative samples to total positive samples.

$$FNR = \frac{FN}{TP+FN} \tag{37}$$

6) False Positive Rate (FPR) is rate of false positive samples to predicted positive samples.

where TP is the true positives; FP is the false positives; TN is the true negatives; and FN is the false negatives.

9. IF Estimation Based on DNN and CNN

A non-stationary signal is a signal that has a changing frequency content across time. This work relies on FM signals, which are affected by noise (AWGN and $S\alpha SN$). $S\alpha SN$ requires four parameters $(\alpha.\gamma.\beta.\, and\ \mu)$. the most important parameters are tail index (α) and scale of the distribution $(\gamma > 0)$; while the less important parameters are $\beta\ and\ \mu$. Gaussian noise is fixed power, $S\alpha S$ noise is geometric power.

Thermal noise is the primary cause of noise in electronic and communication systems. This noise process (typically additive) occurs due to the random thermal agitation of free electrons caused by an electrical current flowing through a conductor. This type of noise is white, meaning that its power spectral density is nearly equal throughout the frequency spectrum. Therefore, thermal noise-affected communication systems are frequently represented as an additive white Gaussian noise (AWGN) channel.

Geometric SNR (GSNR) is used to determine noise impulsiveness, which is characterized by zero-order statistics. Since all 2nd order moments are infinite, the standard SNR does not apply. Geometric power of $S\alpha S$ is defined as follows:

$$p_S = \gamma^2 .\ C^{\left(\frac{2}{\alpha}-1\right)} \tag{38}$$

C is the exponential of Euler's constant, $C = e^{Ec} \approx 1.7811$, Ec is Euler's constant ($Ec = 0.5772156649$). When $\alpha = 2$, SαS noise is Gaussian noise with finite variance $\sigma^2 = \gamma^2$.

$$\text{GSNR} = p_x / \gamma^2 \quad (39)$$

$$\text{GSNRdB} = 10 \times \log_{10}(\text{GSNR}) \quad (40)$$

$$p_x = A^2/2 \quad (41)$$

The received signals in wireless networks are corrupted by a noise that is a mixture of both Gaussian ($\mathrm{N_G}$) and SαS ($\mathrm{N_S}$) noises. Total noise ($\mathrm{N_T}$) represented in the equation:

$$N_T = N_G + N_S \quad (42)$$

Overall GSNR is defined as follows,

$$FNR = \frac{FN}{TP+FN} \quad (43)$$

Let $p_T = p_G + p_S$, where $\mathrm{p_T}$ total noise power, and p_G be the Gaussian power. If $p_G = b \times p_S$, then $p_T = (1+b) \times p_S$, $p_S = \frac{p_T}{1+b}$, and

$$b = p_G / p_S \quad (44)$$

If b less than one, then p_G less than p_S, else p_G greater than p_S. The scale parameter is:

$$\gamma = \sqrt{p_S / C^{\left(\frac{2}{\alpha}-1\right)}} \quad (45)$$

Consider AWG and $S\alpha S$ noise affected by a single-tone sinusoidal and FM signals as follows:

$$x(t) = A.\cos(\varphi + \emptyset_0) + N_T \quad (46)$$

where A signal amplitude, φ is an instantaneous phase, and $\emptyset_0$ initial phase. In this work, dataset generates FM noisy signals with frequencies and slopes different. It also uses Geometric SNR range $sr \in [-50 \;\; 50]$ dB. Dataset are included three types of FM signals are single tone sinusoidal, LFM, and QFM signals. Frequencies values are [10 19]; while slopes values are [0.1: 0.2: 1.0]. Frequencies and slopes estimation or classification by ANN and CNN are as follows:

ANN Model: We used multi layered ANN. Its structure is input layer, two hidden layers, and output layer. Number of nodes in input layer is 101 nodes. Number of nodes in first and second hidden layers are 3 & 3 nodes that use Log sigmoid as transfer function. Number of nodes in output layer is 10 nodes, for frequencies classification and slopes classification. In output layer positive linear transfer function is used as shown in Fig. (2). Fig. (1) shows ANN architecture for frequencies and slopes classification. Number of epoch equals 100. Scaled conjugate gradient optimization algorithm is used for update parameters. All samples in dataset are used for training with GSNR $\in [-50 : 2: 50]$, and generate new samples for test with GSNR $\in [-50 : 2 : 50]$. Number of input samples equals 2550, where each sample length equals 101. Estimating frequency of range is [10 19], and slope of range is [0.1: 0.2: 0.9] using classical methods taking a lot of complexity, where a multi-layer ANN is designed to estimate frequency and slope with less complexity and higher efficiency.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 August 2022 doi:10.20944/preprints202208.0068.v1

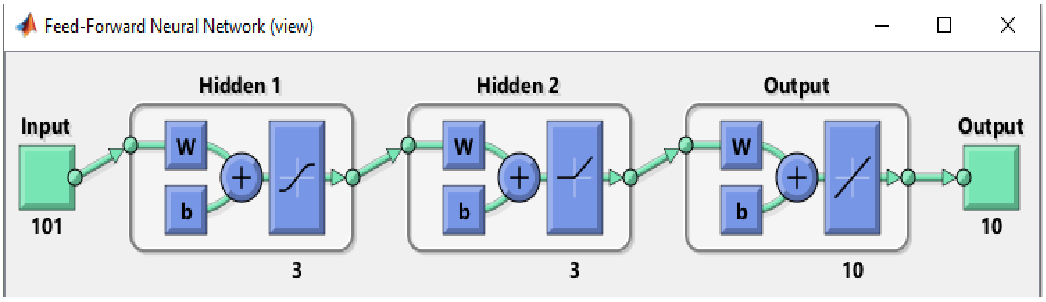

Figure 2. ANN architecture for frequencies classification.

Figure 3. A- positive Linear transfer function; B- Log sigmoid as transfer function.

CNN Model: It deals with 2-dimensional images, so after generating noisy signals, they are converted into 2-dimensional images. The proposed CNN model runs on given a particular set of sample data, which divides data into designing and test, starts training a CNN by splitting the designing set into two sets one set is used for training the CNN and the other one is used as a validation set for testing the generalization ability of the network during the learning process and storing the configuration of the weight that performs best on it with minimum validation error. The aim of the split designing data to training and validation is to reserve a part of the designing data and uses it to monitor the performance. Our samples are divided into 90% for design and 10% for testing, then design data are divided into 90% for training and 10% for validation. Number of samples is 5100, they have 10 classes represent frequency [10 19] Hz & slopes [0.1: 0.1: 1.0]; each class has 510 samples. Fig. (3) shows proposed CNN model layers.

The training procedure is performed by using the backpropagation algorithm and Adam, with the mini-batch equals 5 where each set of the training data is divided into mini-batches and the training errors are calculated upon each mini-batch in the Softmax layer and get backpropagation to the lower layers. The number of epochs is ten. Finally, after the training procedure is finished, the testing set is used to measure the efficiency of the final. The main steps of the proposed training methodology can be summarized as follows:

1. Splitting the database into three sets: training, validation, and testing.
2. Determining the parameter and the architecture of CNN.
3. Training a CNN using a training set.
4. Evaluating the training CNN using the validation set.
5. For N epochs, do steps 3 to 4.
6. Selecting the best CNN with minimal error on the validation set.
7. Evaluating the selection CNN using the test set.

Architecture CNN has many layers that are used for feature extraction and classification, Fig. (4) shows feature extraction and classification for proposed CNN model. CNN model contains 19 layers, the input layer is the first layer which defines the input dimensions, where input image size is 80-by-80, then there is batch normalization layer. The middle layers consist of four convolutional layers, four rectified linear units (ReLU)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 August 2022 doi:10.20944/preprints202208.0068.v1

layers, and four max-pooling layers, there are dropout, two fully-connected layer, Softmax and classification layers. Table (1) shows the topology of the proposed CNN model.

ReLU activation functions: It is often used with convolutional and fully connected layers to add non-linearity for the network. In addition to that, it results in neural network training several times faster than other activation functions. Dropout method: the dropout is often used to remove overfitting in the fully connected (FC) layers. This layer helps prevent all neurons from meeting the same target and speed up the training process.

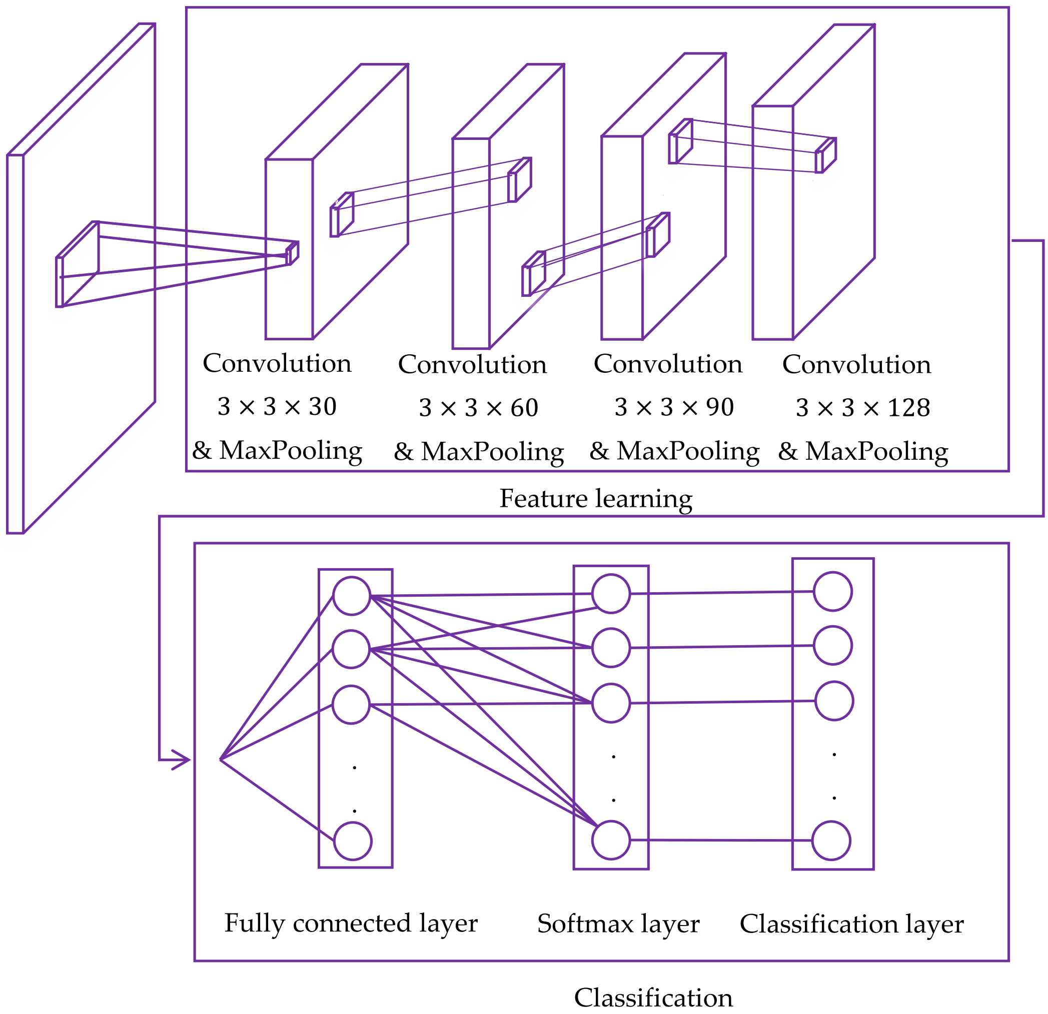

Figure 4. Proposed CNN model layers.

Table 1. Topology of the proposed CNN model.

Indexes	Layers Name	Kernels Size	Stride	Padding
1.	Image input	80 × 80	-	-
2.	Convolution	3 × 3 × 30	1	1
3.	Batch Normalization	-	-	-
4.	ReLU	-	-	-

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 August 2022 doi:10.20944/preprints202208.0068.v1

5.	Max Pooling	2×2	1	-
6.	Convolution	$3 \times 3 \times 60$	1	1
7.	ReLU	-	-	-
8.	Max Pooling	2×2	1	-
9.	Convolution	$3 \times 3 \times 90$	1	1
10.	ReLU	-	-	-
11.	Max Pooling	2×2	1	-
12.	Convolution	$3 \times 3 \times 128$	1	1
13.	ReLU	-	-	-
14.	Max Pooling	2×2	1	-
15.	Fully-Connected	-	-	-
16.	Dropout	50%	-	-
17.	Fully-Connected	-	-	-
18.	Softmax	-	-	-
19.	Classification	-	-	-

10. IF Estimation Based on TFD

The Fourier Transform (FT) cannot detect the time-varying characteristics of non-stationary signals with time-varying frequency content (such as FM signals and biological signals). This is because the FT employs a time-averaging process (time integration). Time-Frequency Distribution (TFD) are two-dimensional double transforms from the time domain to the time-frequency domain representing the Fourier transform of the instantaneous autocorrelation of an analytical signal. The Short-Time Fourier Transform (STFT), a windowed frequency distribution, is the simplest formula for a time-frequency distribution [23, 44-45]. A non-stationary signal is a signal that has a changing frequency content across time. A non-stationary signal's spectrogram provides an estimate of the time evolution of its frequency content. IF estimation by TFD, where FM signals are affected by noise (AWGN and $S\alpha SN$). Estimation of the IF for analytical signals using TFD and STFT. Prior to estimate, employed Hilbert transformation to obtain the analytic signal linked with the noise signal. First, we find spectrogram of STFT (spec$(t.f)$). Then estimate the IF from the peak (max) of the spec as follows:

$$\hat{f}_i(t) = \arg(\max\{\mathrm{spec}(t,f)\});\, 0 \le f \le \frac{f_s}{2} \tag{47}$$

Then, we calculate the relative squared error for each GSNR as follows:

$$e = \left|\left(\hat{f}_i \times df - \mathrm{IF}_t\right)/f_o\right|^2 \tag{48}$$

where f_o fundamental frequency, $\hat{f}_i$ estimated frequency, IF_t theoretical IF with spectrogram timing, $df = \frac{f_s}{N}$ and $N = 1024$. Used spectrogram and pspectrum MATLAB function for IF estimation by TFD; Pspectrum is different from spectrogram in segment lengths, overlapping segments, and window. Spectrogram length $= 1 \times \left[\frac{N}{2} + 1\right]$. pspectrum length $= 1 \times N$. Pspectrum used time resolution and overlap percent pair arguments to control the length of the segments and the overlap between adjacent segments; it is dividing the signal into overlapping segments, applying a Kaiser window to each segment.

11. Discussion of the Results

This section simulates the estimation of the instantaneous frequency and slope of single-tone and FM signals with additive white Gaussian noise and symmetric stable noise. Geometric SNR range is ∈ [-50: 2: 50] dB. Number of samples is (5100) samples with frequencies [10 19] & slopes [0.1: 0.1: 1.0]. This data is divided 10 classes each class is of 510 samples. The network trains the on-input data and predict the frequency once and the slope again. The simulation of frequency and slope estimation for single tone frequency and LFM signals by ANN and CNN model. The results show high accuracy for parameters estimation by confusion matrix and some measures such as (accuracy, precision, F1-score, FNR, & FPR), also few errors rate, and SαS is impulsive model, alpha is more harmful even if it is of small value, where it effects the slope and frequency guess. The ratio between AWGN & SαS is determined by a variable b. Fig. (4) show α-stable probability density functions with different parameters. Fig. (5) show Alpha-Stable noise in time domain. It is impulsive. Fig. (6) show a single tone and noise signals.

Figs. (7-8) show Frequency Estimation (FE) of single tone, and LFM signals by ANN. Figs. (9) Slope Estimation (SE) of single tone, and LFM signals by ANN. Figs. (10-12) show accuracy and loss rate of FE and SE for noisy LFM. Figs. (11-13) show confusion matrix for FE and SE for noisy LFM. Tab. (2) show performance evaluation criteria of noisy LFM signals. Fig. (14) show accuracy of frequency estimation of LFM. Figs. (15) show test error of frequency estimation for LFM. Fig. (16) show accuracy of slope estimation of LFM. Fig. (17) show test error of slope estimation for LFM. Figs. (18-19): show MSE versus GSNR for TFD of noise single tone signal by spectrogram and pspectrum, where α=1 and b=20. Figs. (20-21): show MSE versus GSNR for TFD of noise LFM signal by spectrogram and pspectrum, where α=1 and b=20. Figs. (22-23): show accuracy of FE for noise single tone, and LFM signals by TFD (Pspectrum). Fig. (24) show accuracy of FE for noise LFM by DNN & TFD (spectrogram & Pspectrum). Fig. (25) show error test of FE for noise LFM by DNN & TFD (spectrogram & Pspectrum). Fig. (26): show test error rate for SE of noise LFM by DCNN, where $fo = 19.0005$.

The results showed that artificial neural networks are better than time-frequency distribution for estimating the instantaneous frequency, and deep CNN is better than artificial neural networks in estimating the instantaneous frequency of non-stationary signals. In time-frequency distribute, spectrogram and pspectrum used, where the results showed that pspectrum is better than spectrogram for IF estimate.

12. Conclusion

This paper has presented overall description of the performance of a machine learning and deep-learning approaches for the frequency and slope estimation of a noisy Linear Frequency-Modulated (LFM) and single-tone sinusoid signal. The simulate is a relevant signal under additive white Gaussian noise and symmetric stable noise (impulsive model). Geometric SNR range is ∈ [-50 50] dB. This work processes the problems in classical approaches, it relies on machine learning and deep learning. It views analysis of the frequency and slope estimation error under a range of Geometric signal to noise ratios (GSNRs). In ANN, few hidden layers only two are used. They include ten nodes in first hidden layer and eight nodes in second hidden layer. In CNN model 14 layers are used, where three convolution layers are used with three ReLU activation functions two Max-pooling layers, dropout layer, two fully connect layer, Softmax layer, and classification layer relies on cross entropy to find cost function. The simple structure designed for ANN or CNN model works on reducing the complexity, power consumption, and cost of the communication system. These characteristics are beneficial for systems with finite memory and computational processes, such as Wireless Sensor Networks (WSNs), that connect with applications on the Internet of Things. The simulation result shows that alpha is more harmful even if it has small incapacity, and it has a significant effect on guess frequency and slope. The ratio between AWGN & SαS is determined by a variable (b).

doi:10.20944/preprints202208.0068.v1

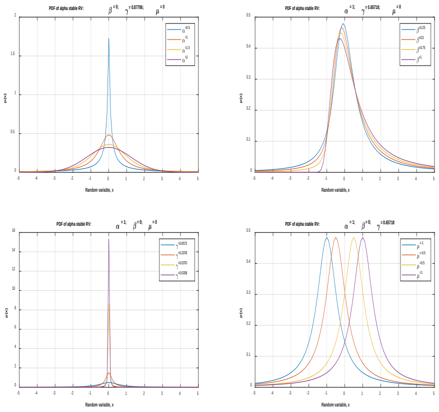

Figure 4. α-stable probability density functions with different parameters.

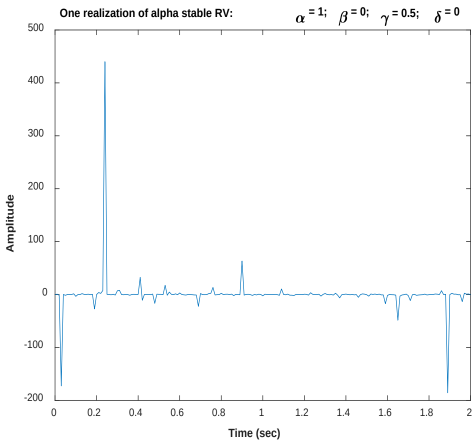

Figure 5. α -Stable noise in time domain. It is impulsive.

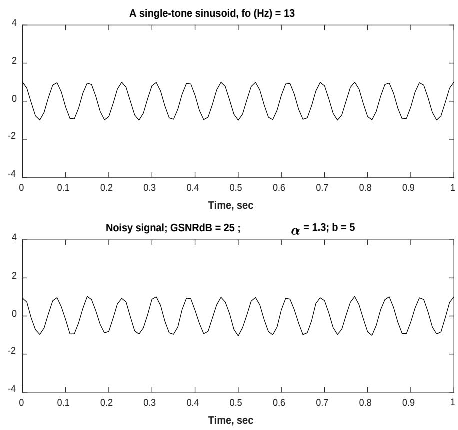

Figure 6. A single tone and noise signals.

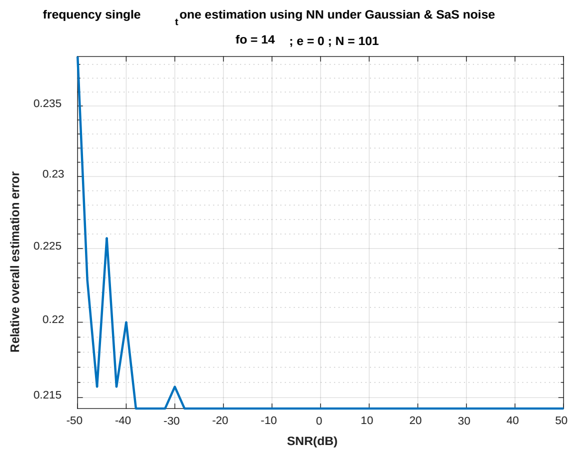

Figure 7. FE of noisy single tone signals by ANN.

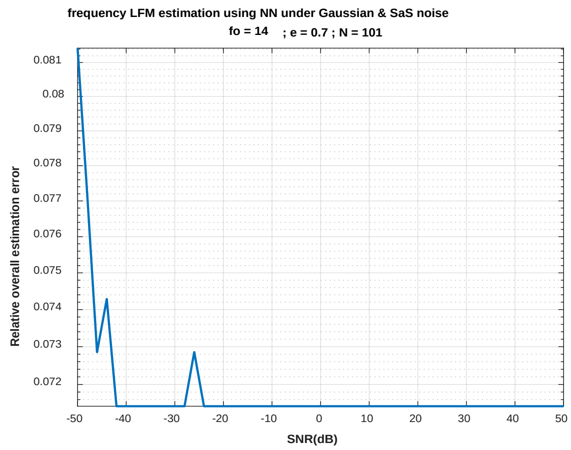

Figure 8. Error rate for FE of noisy LFM signals by ANN.

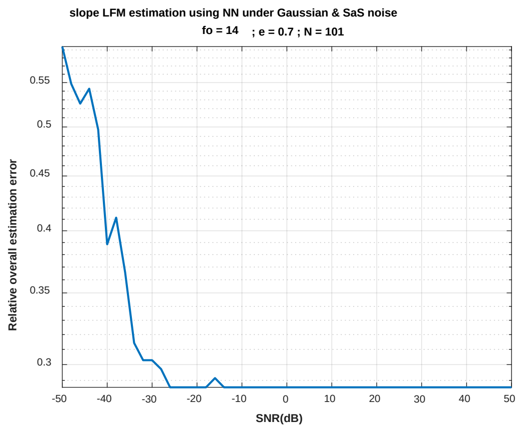

Figure 9. Error rate for SE of noisy LFM signals by ANN.

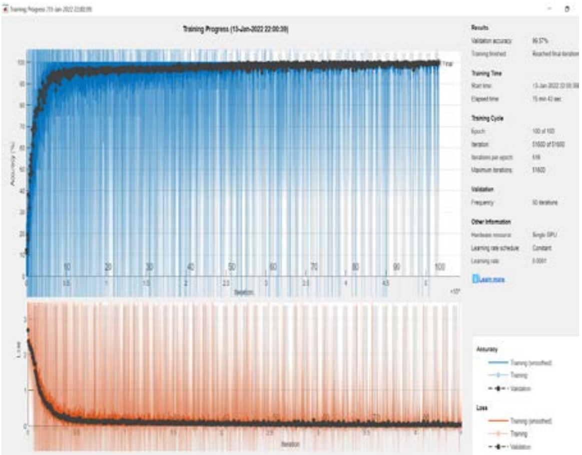

Figure 10. Accuracy and loss rate of FE for noisy LFM by DCNN.

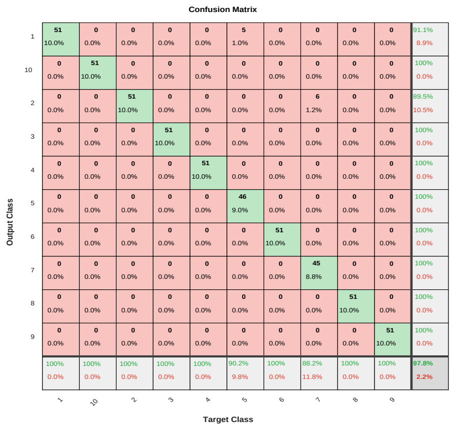

Figure 13. A confusion matrix of SE for LFM.

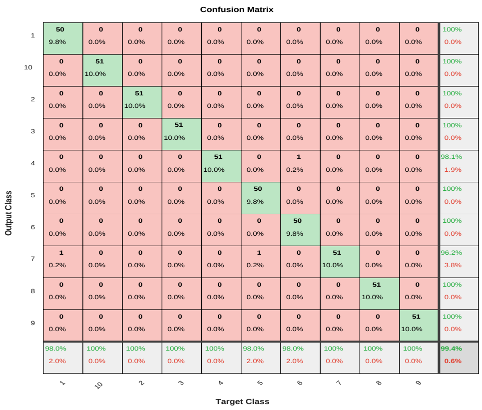

Figure 11. A confusion matrix of FE for LFM signals by DCNN.

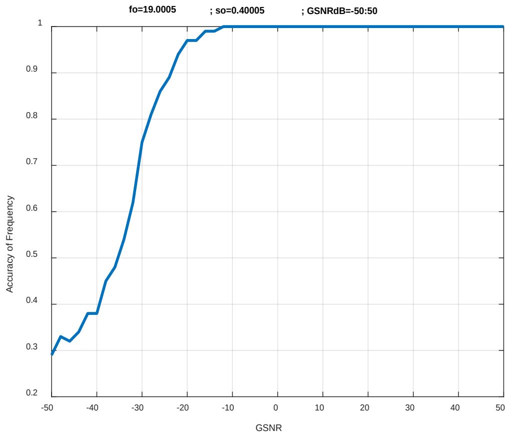

Figure 14. Accuracy of FE for noise LFM by DCNN, where f_o=19.0005.

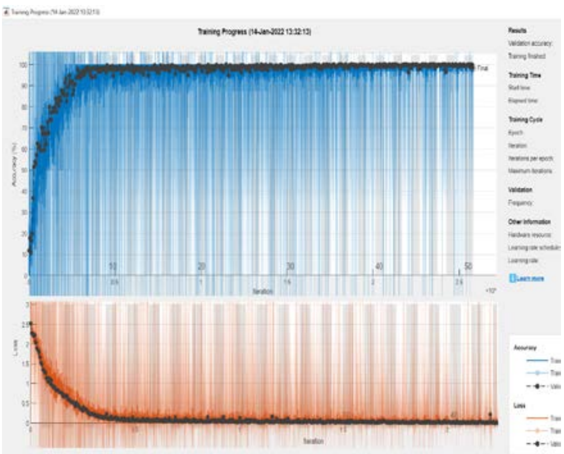

Figure 12. Accuracy and loss rate of SE for noisy LFM by DCNN.

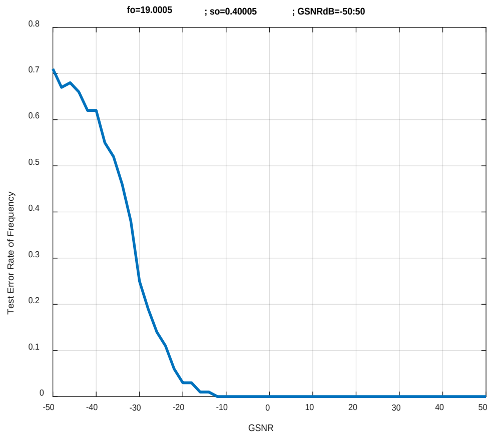

Figure 15. Test error rate of FE for noise LFM by DCNN, where f_o=19.0005.

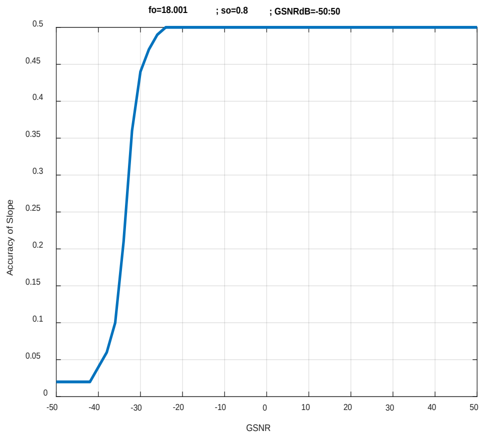

Figure 16. Accuracy of SE for noise LFM by DCNN, where f_o=18.001.

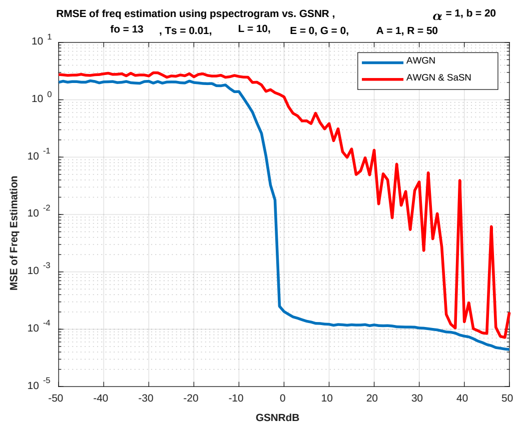

Figure 19. MSE versus GSNR for TFD of noise single tone signal by pspectrum, where α=1 and b=20.

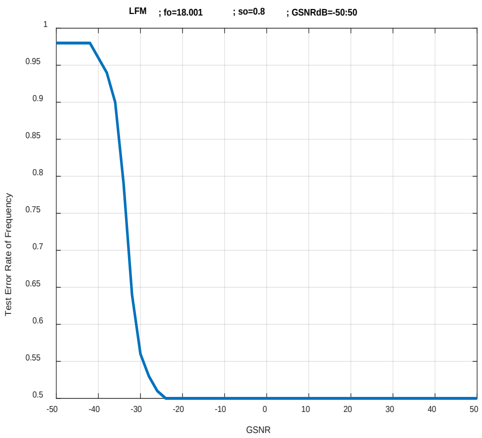

Figure 17. Test error rate for SE of noise LFM by DCNN, where f_o=18.001.

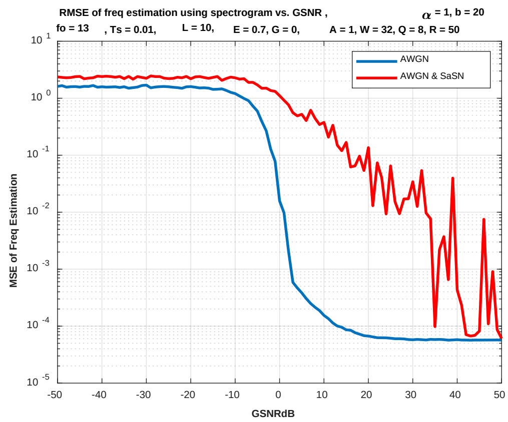

Figure 20. MSE versus GSNR for TFD of noise LFM signal by spectrogram, where α=1 and b=2.

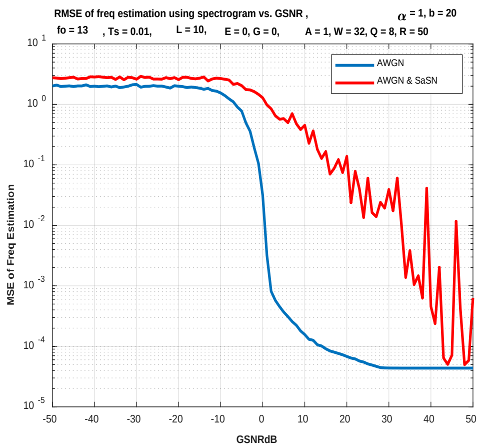

Figure 18. MSE versus GSNR for TFD of noise single tone signal by spectrogram, where α=1& b=20.

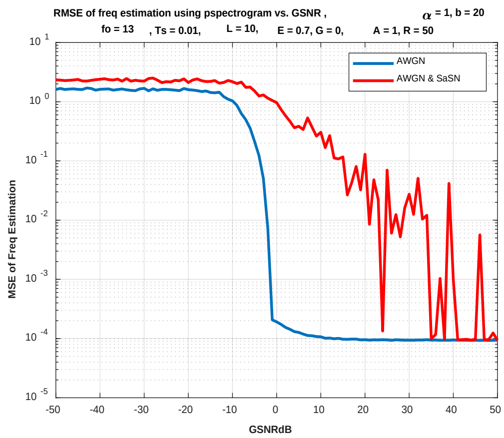

Figure 21. MSE versus GSNR for TFD of noise LFM signal by pspectrum, where α=1 and b=20.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 August 2022 doi:10.20944/preprints202208.0068.v1

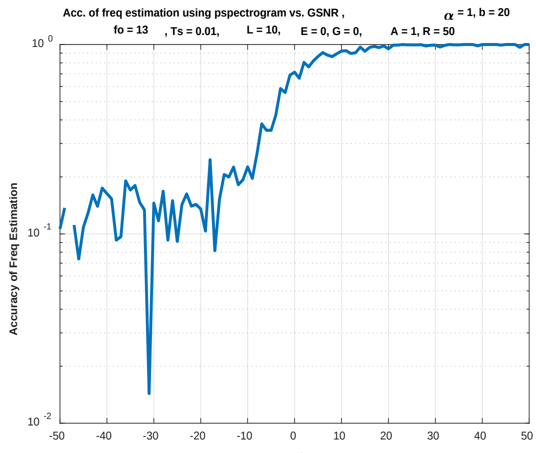

Figure 22. Accuracy of FE for noise single tone signal by TFD (Pspectrum).

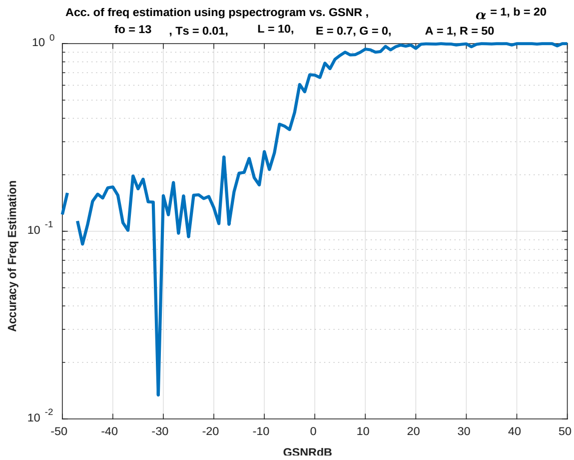

Figure 23. Accuracy of FE for noisy LFM by TFD (Pspectrum).

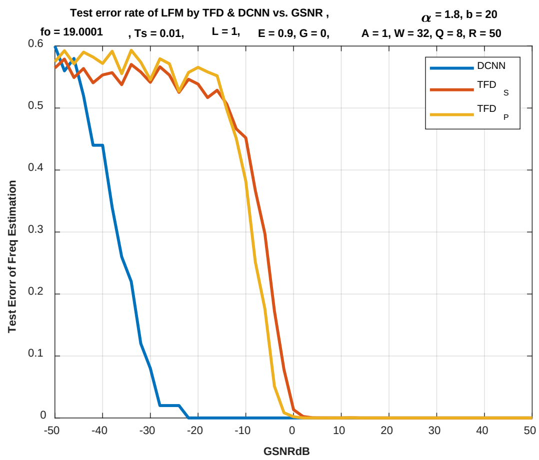

Figure 24. Accuracy of FE for noise LFM by DNN and TFD for spectrogram (TFD_s) & Pspectrum (TFD_p).

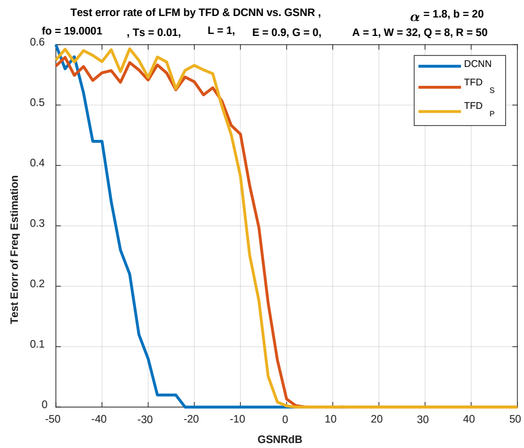

Figure 25. Error rate of FE for noisy LFM by DNN and TFD for spectrogram (TFD_s) & Pspectrum (TFD_p)

Table 1. Measures of FE & SE for noisy LFM signals.

Measures	LFM	
	Frequency	**Slope**
Accuracy	99.4118	97.8431
Precision	99.4303	98.0545
Recall	99.4118	97.8431
F1_Score	99.4210	97.9487
FNR	0.0059	0.0216
FPR	0.0057	0.0195

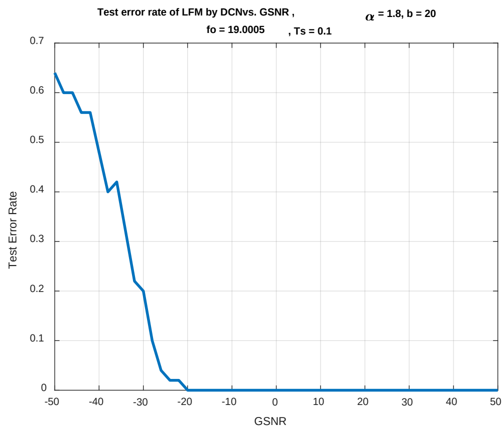

Figure 26. Test error rate for SE of noise LFM by DCNN, where f_o=19.0005.

Author Contributions: H.S.R. and Z.M.H. contributed equally to this work. H.S.R. contributed to the initial tasks of this work during her post-graduate preparatory year of coursework in 2019, where she had obtained the top rank of high-distinction before she started her research project on deep learning for frequency estimation under the supervision of Z.M.H. All authors have read and agreed to the published version of the manuscript.

Funding: This project was partially funded by Edith Cowan University via the ASPIRE Program.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All types of data were generated using mathematical equations.

Acknowledgments: The authors would like to thank Edith Cowan University for partially supporting this project.

Conflicts of Interest: The authors declare that no conflict of interest is associated with this work.

References

1. Boashash B., "Estimating and Interpreting the Instantaneous Frequency of a Signal. I. Fundamentals," Proceedings of the IEEE, 1992.
2. Boashash B., "Estimating and Interpreting the Instantaneous Frequency of a Signal. II. Algorithms and Applications," Proceedings of the IEEE, 1992.
3. Liu Hongyu and Bo Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey", applied sciences, 2019.
4. Boashash B., "Estimating and Interpreting the Instantaneous Frequency of a Signal. I. Fundamentals," Proceedings of the IEEE, 1992.
5. Boashash B., "Estimating and Interpreting the Instantaneous Frequency of a Signal. II. Algorithms and Applications," Proceedings of the IEEE, 1992.
6. Liu Hongyu and Bo Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey", applied sciences, 2019.
7. Ghorbanzadeh O., Blaschke T., Gholamnia K., MeenaS. R., Tiede D., & Aryal J., "Evaluation of different machine learning methods and deep-learning convolutional neural networks for landslide detection"., Remote Sensing, 11(2), 196, 2019.
8. Tang T. A., Mhamdi L., McLernon D., Zaidi S. A. R., Ghogho M., & El Moussa F., "DeepIDS: deep learning approach for intrusion detection in software defined networking". Electronics, 9(9), 1533, 2020.
9. Woschank M., Rauch E., & Zsifkovits H., "A review of further directions for artificial intelligence, machine learning, and deep learning in smart logistics". Sustainability, 12(9), 3760, 2020.
10. Yao Y., Hu W., Zhang W., Wu T., & Shi Y. Q.," Distinguishing computer-generated graphics from natural images based on sensor pattern noise and deep learning". Sensors, 18(4), 1296, 2018.
11. Zantalis F., Koulouras G., Karabetsos S., & Kandris D., "A review of machine learning and IoT in smart transportation", Future Internet, 11(4), 94, 2019.
12. Liu J., Fan L., Jin J., Wang X., Xing J., & He W., "An Accurate and Efficient Frequency Estimation Algorithm by Using FFT and DTFT," In 39th Chinese Control Conference (CCC), IEEE, 2020.
13. Akram J., Khan N. A., Ali S., & Akram, A. "Multi-component instantaneous frequency estimation using signal decomposition and time-frequency filtering," Signal, Image and Video Processing 14, 2020.
14. Xu S., & Shimodaira H., "Direct F0 Estimation with Neural-Network-Based Regression", In INTERSPEECH (pp. 1995-1999), 2019.
15. Silva Bruno, et al. "Artificial Neural Networks to Solve Doppler Ambiguities in Pulsed Radars." 2018 International Conference on Radar (RADAR). IEEE, 2018.

16. Chen X., Jiang Q., Su N., Chen B., & Guan J., " LFM Signal Detection and Estimation Based on Deep Convolutional Neural Network". In 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) (pp. 753-758). IEEE, 2019.
17. Liu Xuelian, and Chunyang Wang. "A novel parameter estimation of chirp signal in α-stable noise," IEICE Electronics Express, 2017.
18. Milczarek, Hubert, et al. "Estimating the Instantaneous Frequency of Linear and
19. Nonlinear Frequency Modulated Radar Signals—A Comparative Study," Sensors 21.8, 2021
20. 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.
21. B. Boashash, P. O'Shea, M. J. Arnold, "Algorithms for instantaneous frequency estimation: a comparative study," Proc. SPIE, 1990.
22. 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.
23. B. Boashash, Ed., “Time-Frequency Signal Analysis and Processing: a comprehensive Reference”, Elsevier, Oxford, UK, 2016.
24. 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.
25. 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.
26. Bengio Y.,”Learning deep architectures for AI”. Foundations and trends® in Machine Learning, 2(1), 1-127, 2009.
27. Cristianini N., & Shawe-Taylor, J., “An introduction to support vector machines and other kernel-based learning methods”. Cambridge university press, 2009.
28. Yager R. R., “An extension of the naive Bayesian classifier”. Information Sciences, 176(5), 577-588, 2006.
29. Guo G., Wang H., Bell D., Bi Y., & Greer K. , “KNN model-based approach in classification”. In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems" (pp. 986-996). Springer, Berlin, Heidelberg, 2003.
30. Safavian S. Rasoul, and David Landgrebe. "A survey of decision tree classifier methodology." IEEE transactions on systems, man, and cybernetics 21.3, 1991.
31. Alpaydin E., “Introduction to machine learning”. MIT press, 2009.
32. Géron A., “Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems. " O'Reilly Media, Inc.", 2017.
33. J. Shlens, “A Tutorial on Principal Component Analysis,” 2014.
34. C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation Functions: Comparison of trends in Practice and Research for Deep Learning,” pp. 1–20, 2018.
35. Phil K., "Matlab deep learning with machine learning, neural networks and artificial intelligence", Apress, New York, 2017.
36. D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” pp. 1–15, 2014.
37. LeCun Y., Bengio Y., Hinton G., “Deep learning”, Nature 2015, 521, 436.
38. 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.
39. Gad, Ahmed Fawzy, "Practical Computer Vision Applications Using Deep Learning with CNNs", 2018.
40. Moons B., Bankman D., & Verhelst M., “Embedded Deep Learning: Algorithms, Architectures and Circuits for Always-on Neural Network Processing”, Springer, 2018.
41. Srivastava N., Hinton G., Krizhevsky A., Sutskever I., & Salakhutdinov R., “Dropout: a simple way to prevent neural networks from overfitting”, The journal of machine learning research, 15(1), 1929-1958, 2014.
42. Powers D. M., “Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation”, 2011.
43. Amin Vaishali S., Yimin D. Zhang, and Braham Himed. "Improved instantaneous frequency estimation of multi-component FM signals.”, IEEE Radar Conference (RadarConf). IEEE, 2019.

doi:10.20944/preprints202208.0068.v1

44. Hussain, Z. M., Sadik A. Z., and O'Shea P., Digital Signal Processing, Springer, Berlin, Germany, 2011.
45. Huda Saleem Razzaq, 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.