**EE305 Design Laboratory**

Deep Learning for Improving Performance of

OOK Modulation Over FSO Turbulent Channels



*A design lab report*

*submitted by*

## Akshat Porwal (1901EE09)

## Omkar Deshpande (1901EE43)

## Shubham Kumar Diwakar (1901EE58)

**Abstract**

Due to its low power consumption, high speed and high capacity, Free Space Optical (FSO) communication technology is recently gaining traction. The only hindrance in mass use of this technology is it struggles in turbulent atmosphere. Turbulent atmosphere causes scintillations, regardless of the modulation used.

This report presents and compares 3 Deep Learning models for improving performance of OOK modulation over FSO channels. The 3 models are based on Artificial Neural Networks (ANN), Fully Connected Layers (FC) and Fully Convolutional Neural Networks (FCNN) respectively. The comparison among the 3 models is based on accuracy and time complexity.

**Introduction**

Free space optical (FSO) communication has gained significant attention due to its high bandwidth and data rate capabilities (as good as optical fibres) which can support the rapid growth of different cloud applications such as internet and cell phones. In FSO, the data is transmitted via light over a FSO channel. The signal transmission occurs at line of sight. (LOS). As a result, FSO can create networks which are more flexible and robust as compared to pre-existing methods such as optical fibres and Radio Frequency (RF) systems. The infrastructure of FSO is easier and cheaper to lay down as compared to optical fibre and unlike RF systems, no spectrum licence is required since data is transferred over line of sight (LOS).

However, the main factor preventing mass use of FSO communication is turbulence. Since the ray of light carrying all the data travels through a turbulent FSO channel, there are unexpected changes in the amplitude and phase of the signal. The turbulence is created due to the constantly changing refractive index of air in the atmosphere. To overcome this, one may need exact details of the channels beforehand, that is channel state information (CSI).

We compare the Deep Learning Models with the existing decoding methods of the OOK encoding - Maximum Likelihood (ML) with perfect CSI decoder and traditional OOK decoder with fixed threshold. Later we will see that the Deep Learning Models work best among other methods.

**Generating Input :-**

In order to check the performance of our suggested DL models

We need to generate sets of input and output data, train and test our models, and then compare between the performance of our models and the OOK(ON-OFF Keying) with a threshold detection method. For this, using MATLAB software we generated a dataset of 39000 samples of random data bits, each one with a size of 512 bits.

For each 26 values of SNR from 5 to 30, we generate 500 samples of 512 bits and this process is repeated for each type of atmospheric turbulence.

Here we have considered 3 types of atmospheric turbulence :- namely Low , Moderate , and Strong Turbulence . We modulated each vector of data by OOK modulation and transmitted it across different strength FSO turbulence channels with AWGN( Additive White Gaussian Noise).We also store the random data bits vector. The input of our DL models is the received modulated data with noise that arrived at the receiver, and the output is the original data bits that we generated.

Here we have used 90% data generated for training and remaining 10% data for testing.

**Scaling Input Data:-**

We need to scale the input data as there can be outliers which can negatively affect our results/Predictions. So we need to normalise the data such that the generated signal values are between [-1,1].

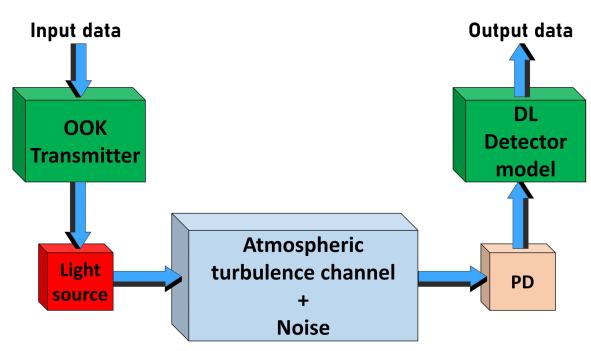
Let the Current value be , mean value of modulated vector is ,

Maximum value of modulated vector is ,Minimum value of modulated vector is then we can replace the value of modulated vector’s current bit by :

.

**FSO turbulence channel**

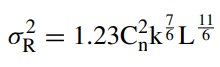
FSO communication is an optical communication technology that transmits data via light through free space using intensity modulation (IM). The transmitted data propagates through a turbulent channel with additive white Gaussian noise (AWGN). At the receiver, the data is received via a photodetector (PD) and is detected using direct detection (DD), as in the figure below.



We assume that the channel is memoryless and stationary, and exhibits slow fading. The received signal can be described by the basic channel model

= ɳh + noise

where η is the responsivity of the PD (measured in V/W), h is the channel state, which includes attenuation due to atmospheric turbulence, and is equal to the channel intensity at that time. Above equation characterises the received data affected by distortions caused by turbulence. When an FSO signal travels through air, it is subject to fading and degradation. Rytov variance, is the parameter that determines the type of turbulence and it can be calculated according to:

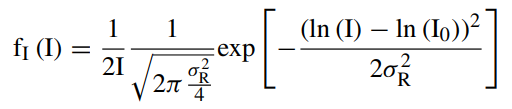


where is the refractive index structure coefficient that describes the

fluctuations and changes in air temperature through the chan-

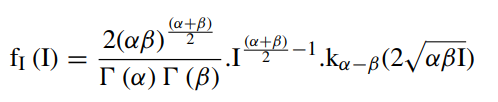
nel.

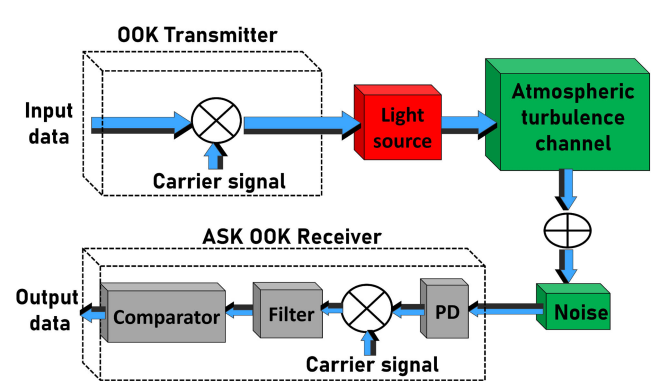
When σ2R << 1, the turbulence is weak. Otherwise the turbulence is strong. For modelling weak turbulence, we use a lognormal distribution with a PDF :

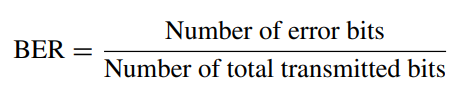


where I is the received signal intensity, σR2 is the variance of the log amplitude of the received signal, and ln (I0) is the average log intensity of the received signal.

For modelling strong turbulence we use Gamma Gamma distribution with a PDF :



OOK modulation is On Off Keying modulation. It is used widely in FSO systems majorly due to its simplicity. In this modulation technique, a bit ‘‘one’’ is modulated by the carrier frequency and represented by an optical pulse. When the bit is zero the transmitter is in mode ‘‘off’’ and, in this time interval, the transmitter is not active

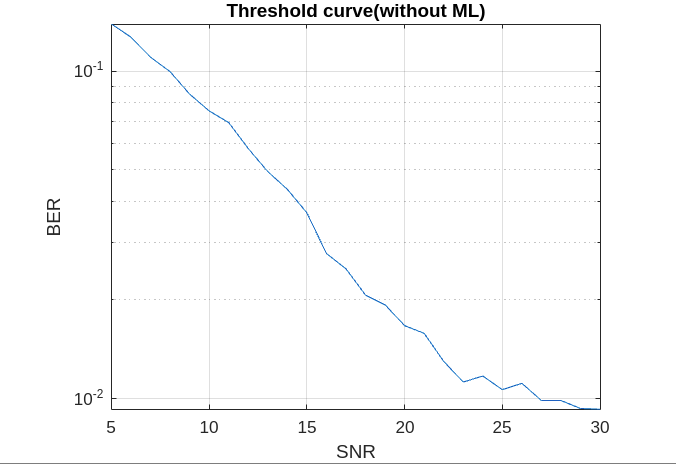
SNR is Signal to Noise Ratio and BER is the Bit Error Ratio. BER is our measure of accuracy of the received signal. In order to achieve lower bounds of BER and good performance in the case of higher values of the scintillation index parameter, the SNR is increased and more power is transmitted and in some cases, too much power is consumed. The existing reception methods like ML decoders require perfect CSI and are hence not able to cope with ever changing conditions of the atmosphere, hence there is a high demand to solve these problems.

**Proposed DL Detection Models For FSO**

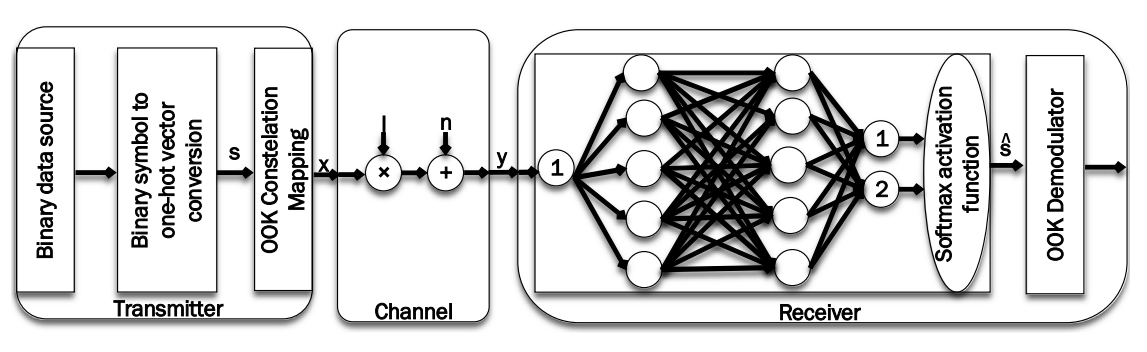
**Fixed Threshold based Model**

The detection of the received data bits in this model is calculated according to the threshold, which is the mean of the input signal. If the value is higher than the threshold then the detected bit is 1, and otherwise it is 0.

Here the input is the randomly generated signal passed through the modulation and turbulence.



**Artificial Neural Network based model**

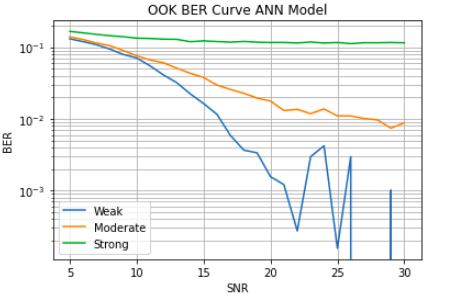


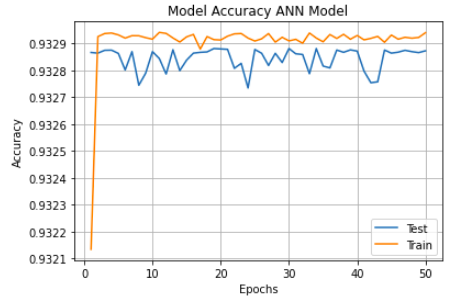
The deep learning based detector enters the received symbol into a ANN and detects it, and does not need any information about SNR, CSI, or anything else for training the ANN. The generated binary symbol is converted to a one-hot vector s ∈ S = {ei | i = 1, 2}, where ei is equal 1 at row i and else 0, then mapped on an OOK constellation. The aim of the training is to reduce the loss, which in our case is the difference between the ANN output (the detected symbol) and the target (the transmitted symbol). Then, the mapped OOK symbol is transmitted by an FSO transmitter through an atmospheric turbulence channel, and added by the AWGN with zero mean and σ2 variance at the FSO receiver. The received electrical signal is then entered into an ANN with Nhid hidden layers, Nneu neurons per hidden layer, some activation functions, weight matrices, and bias vectors. The purpose is to adjust ANN parameters (weight matrix and bias vector) such that the receiver could better recover the transmitted signal.

**Tuned hyperparameters in this model**

| **Hyperparameters** | **Value** |
| --- | --- |
| No. of layers | 2 |
| Number of hidden neurons | 5 |
| Batch size | 256 |
| Sample size/batch size | 4 |
| Number of Iterations | 1000 |
| Activation function | ReLU |
| Loss | Binary Cross-Entropy |
| Optimizer | Adam |
| Learning rate | 0.001 |
| Gamma-Gamma Atmospheric turbulence Intensity | Strong (𝛼 = 4.2, 𝛽 = 1.4)  Moderate (𝛼 = 4, 𝛽 = 1.9)  Weak (𝛼 = 11.6, 𝛽 = 10.1) |

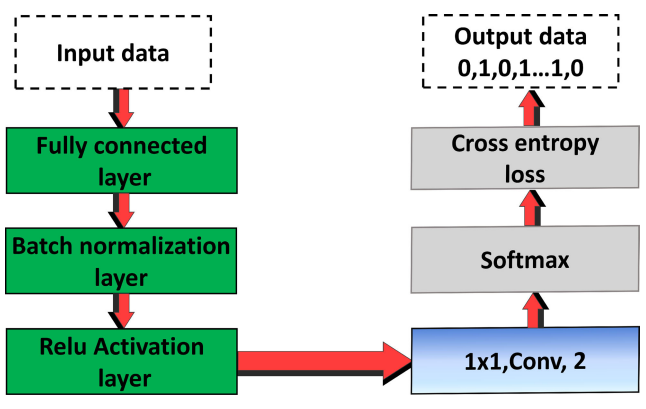
**Results**



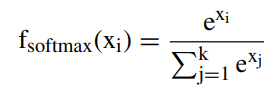


[Link to code](https://colab.research.google.com/drive/1wTexzvY-j_xTSORt-bvYlop39-3Q-pIl#scrollTo=J3D-4XZJ4qn0)

**Fully Connected Network based model**



In our DL models, we used a Relu activation function: fRelu(xi) = max(0, xi) after each internal layer and the last layer is a convolutional layer with two filters of size 1 × 1. This layer is followed by a softmax activation layer:

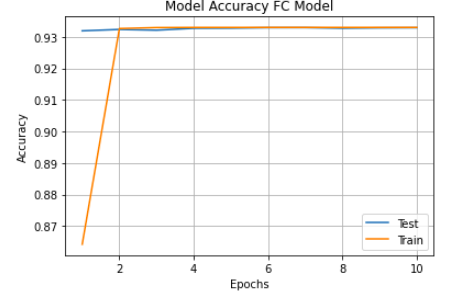
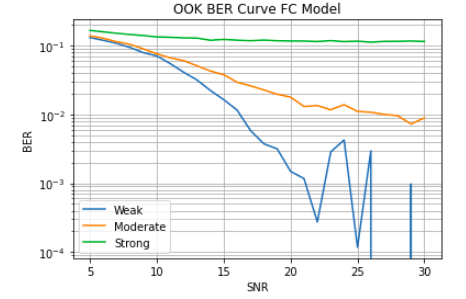
****

that converts the values of the output data from this layer to probabilities from values 0 to 1. At the end we use the cross entropy function to determine the loss, that is the difference in the probabilities, the original ones and the ones predicted by our model. The distance between the output of the DL system after the softmax at the last layer, and the original data bits needs to be minimum by cross-entropy loss. After the FC layer, a convolutional layer was used with 2 filters, each one with a size 1 × 1. At the output of this layer we used a softmax activation function, followed by a binary cross-entropy loss function.

**Tuned hyperparameters in this model**

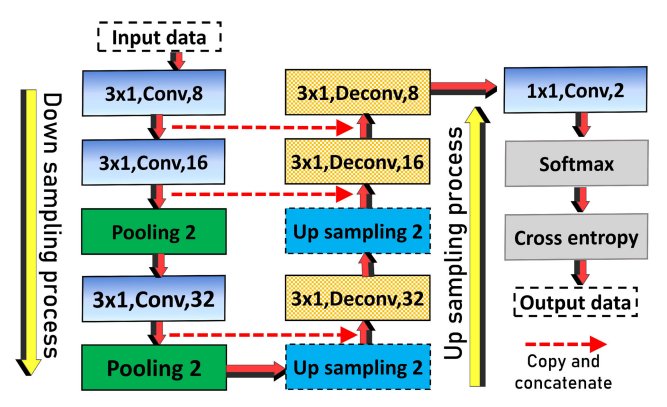
| **Hyperparameters** | **Value** |
| --- | --- |
| No. of layers | 4 |
| Number of hidden neurons | 64 |
| Batch size | 256 |
| Sample size/batch size | 4 |
| Number of Iterations | 1000 |
| Activation function | ReLU |
| Loss | Binary Cross-Entropy |
| Optimizer | Adam |
| Learning rate | 0.005 |
| Gamma-Gamma Atmospheric turbulence Intensity | Strong (𝛼 = 4.2, 𝛽 = 1.4)  Moderate (𝛼 = 4, 𝛽 = 1.9)  Weak (𝛼 = 11.6, 𝛽 = 10.1) |

**Results:-**



[Link to code](https://colab.research.google.com/drive/1FCtZJCsVVKRSmImGa4duuWNCu8uwXGi8#scrollTo=sZe9lV79Zyxb)

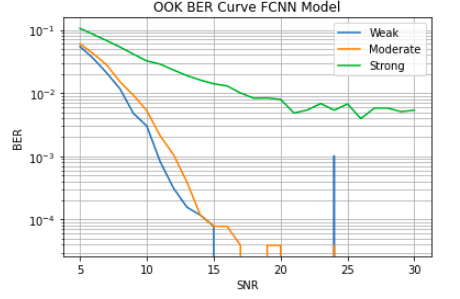
**Fully Convolutional Neural Network based model**

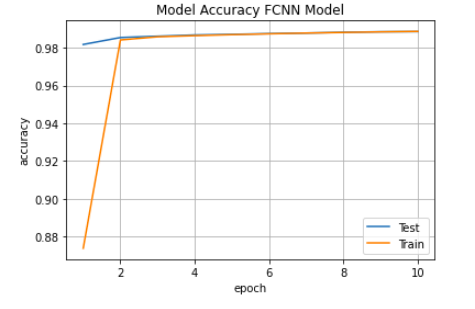


This model includes two process, namely down sampling and up sampling. The down sampling process comprises a number of convolution and pooling layers, and the up sampling process is the inverse process of down sampling, comprising a number of up sampling and deconvolutional layers. The down sampling process allows the model to extract the high level features of the data and recover the information lost due to the convolutional and pooling layers. This also obtains the precise information and localization of the extracted data by an up sampling process. In the down sampling process, a convolutional layer with 8 filters is used, each one with a size of 3 × 1. Then we used another 2 convolutional layers which duplicated the number of the filters to 16 and 32. After each convolution layer, we used pooling layers. In the up sampling process, we used the inverse process that we used before in the down sampling process. After the up-sampling process, we used convolutional layers of two classes followed by a softmax activation function. Then, the cross-entropy loss function is used to calculate the minimum loss between the input and the detected data bits. The model received the same modulated OOK data after passing through turbulence channels with AWGN, and the output of these models is a vector of data bits recovered through the cross-entropy loss function.

**Tuned hyperparameters in this model**

| **Hyperparameters** | **Value** |
| --- | --- |
| Batch size | 256 |
| Sample size/batch size | 4 |
| Number of Iterations | 50 |
| Activation function | ReLU |
| Loss | Binary Cross-Entropy |
| Optimizer | Adam |
| Learning rate | 0.005 |
| Gamma-Gamma Atmospheric turbulence Intensity | Strong (𝛼 = 4.2, 𝛽 = 1.4)  Moderate (𝛼 = 4, 𝛽 = 1.9)  Weak (𝛼 = 11.6, 𝛽 = 10.1) |

**Results**



[Link to code](https://colab.research.google.com/drive/1sckNxAQ_fW8cY4CIh8fjoyWv7R2WVlhn#scrollTo=GbY4oG5mkduB)

**Comparison between the models:**-

Here we have compared the three deep learning models and the fixed threshold model on two parameters: 1) Accuracy vs Epochs 2) BER vs SNR

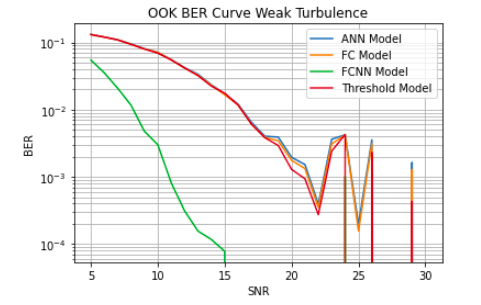
**Accuracy vs Epochs:-**

* In the ANN based model we have achieved an accuracy of 93.29% but we had to use 50 Epochs to achieve that accuracy.
* In the FC based model also we have achieved 93.% but by using only 10 Epochs to achieve that accuracy.
* In the FCNN based model we achieved the accuracy of 98.89%, also we used only 10 Epochs to achieve that accuracy.

**BER vs SNR:-**

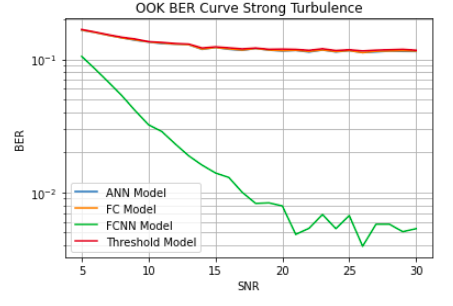
We have plotted BER vs SNR Plot for different turbulences of each type of model.

1. Weak Turbulence:



As we can see, in the FCNN model the lowest BER value is obtained for a particular SNR. For the other three models, nearly the same BER value is obtained for a particular SNR. In those three models also FC model gets the lowest BER then comes the ANN model.

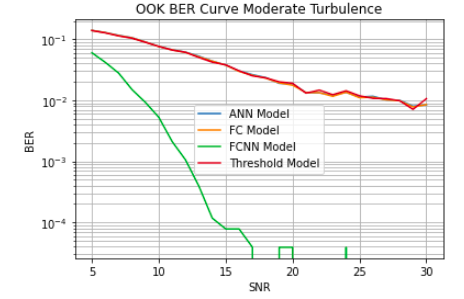
2)Strong Turbulence:



As we can see, in the FCNN model the lowest BER value is obtained for a particular SNR. For the other three models Nearly same BER value is obtained for a particular SNR. In those three models also the threshold model gives the worst output.

Also one thing we can observe is that all models perform better in weak turbulence conditions than the high turbulence conditions.

3)Moderate Turbulence:-



As we can see, in the FCNN model the lowest BER value is obtained for a particular SNR. For the other three models Nearly same BER value is obtained for a particular SNR. In those three models also the threshold model gives the worst output.

Also one thing we can observe is that all models perform better in weak turbulence conditions than the moderate turbulence conditions.Also the results in moderate turbulence case is better than the strong turbulence condition.

So the final conclusion from the results obtained from 3 models and the fixed threshold model is :

* For a particular model its performance is best in weak turbulence followed by moderate turbulence followed by high turbulence conditions.
* For a particular turbulence results are best for FCNN based model followed by FC model followed by ANN model followed by Fixed Threshold model.

**Conclusion:-**

This paper introduces three deep learning based models for combating effects of atmospheric turbulence on the performance of FSO system. We studied variation of bit-error-rate(BER) over signal-noise-ratio(SNR) and three different atmospheric turbulences. The results clearly depict the high performance of FCNN model over all other neural network architectures. In addition, this structure is robust to any changes in the structure of the system, and channel, and by a simple training procedure could get adapted to any structural changes.

**Future work:-**

The FSO fading due to atmospheric turbulence has a good scope of improvement in terms of deep learning based models, like different combinations of CNN layers can be considered with different sizes of kernels. Also our work limits to OOK modulation while we can check the performance of deep learning models for higher modulation schemes as well such as PAM4 and QPSK and QAM.

**Acknowledgement:-**

We are grateful to our guide, Dr. Sumanta Gupta, Associate Professor, Dept. of Electrical Engineering, IIT Patna, for giving us this opportunity and his helpful comments and suggestions.

**References:-**

* Darwesh, L., Kopeika, N.S.: Deep learning for improving performance of OOK modulation over FSO turbulent channels. IEEE Access. 8, 155275– 155284 (2020)
* Amirabadi, M.A., Kahaei, M.H., Nezamalhosseini, S.A.: Deep learning based detection technique for FSO communication systems. Phys. Commun. 43, 101229 (2020)