J. Electrical Systems 20-5s (2024): 423-429

[1] S. Senthil Kumar
[2] Yuvaraja Thangavel
[3] L. Raja
[4] E. Kannan

Machine Learning Algorithms for Performance Enhancement in DCO-OFDM-Based LiFi Systems

Journal of Electrical Systems

Abstract: - This study describes how machine learning methods can be used to enhance the performance of LiFi (light fidelity) systems that are based on DCO-OFDM (direct current bias optical orthogonal frequency division multiplexing). LiFi, a rapidly developing technology that uses light for wireless communication, has to contend with issues like channel impairments and signal interference. This work investigates several machine learning strategies to address these problems and improve the DCO-OFDM LiFi systems' overall performance. It is determined how well various machine learning techniques perform in terms of improving system parameters, decreasing latency, and raising throughput by analyzing real-world data and simulation studies. Various sorts of symmetrical recurrence division multiplexing (OFDM, for example, DC one-sided optical OFDM (DCO-OFDM), are utilized in light devotion (LiFi). A critical DC predisposition in DCO-OFDM brings about optical power failure, while a minor inclination increment cutting commotion. In this manner, it's essential to decide the appropriate DC predisposition level for DCO-OFDM. This examination finds the ideal DC-predisposition an incentive for DCO-OFDM based LiFi frameworks utilizing machine learning (ML) algorithms. To do this, a MATLAB device is utilized to create a dataset for DCO-OFDM. Thusly, Python writing computer programs is utilized to apply machine learning methods. ML is used to recognize the basic DCO-OFDM qualities that influence the ideal DC predisposition. Here, it is exhibited that the base, greatest, and standard deviation of the bipolar OFDM signal, as well as the size of the heavenly body, decide the ideal DC inclination.

Keywords: Machine Learning, Algorithms, DCO-OFDM, LIFI, machine learning (ML)

I. INTRODUCTION

With its ability to transmit data at high speeds using light waves instead of radio frequency (RF), light-fidelity (LiFi) technology has become a viable substitute for conventional RF-based communication systems [1]. Direct Current Offset Quadrature Amplitude Modulation (DCO-OFDM) is one of the modulation methods used in LiFi systems that has drawn a lot of attention because of its great spectral efficiency and resilience to optical channel defects [2]. But like any communication system, DCO-OFDM based LiFi systems have to overcome obstacles to improve and optimize performance, especially in surroundings that are dynamic and heterogeneous [3]. Algorithms for machine learning (ML) have become extremely effective in the last several years in solving complicated optimization issues in a wide range of fields, including wireless communication systems. Researchers and engineers can create clever methods to improve the performance of DCO-OFDM based LiFi systems by utilizing the power of machine learning algorithms [4]. These algorithms can maximize data throughput and reliability by adaptively optimizing system settings, mitigating channel impairments, and enhancing overall system efficiency in real-time [5]. There are various benefits to integrating ML algorithms into DCO-OFDM based LiFi systems. First of all, machine learning (ML) algorithms have the ability to identify patterns and relationships in massive datasets, which allows them to adaptively modify system parameters in response to shifting environmental factors like user movement and changes in ambient light intensity [6]. Second, in order to enhance spectrum efficiency and reduce interference, ML algorithms can optimize the distribution of optical resources, such as LED intensity levels and subcarrier allocation. Thirdly, ML algorithms can improve the robustness and reliability of data transmission in LiFi networks by strengthening error detection and correction processes [7].

There are a number of obstacles in the way of applying ML algorithms for performance improvement in DCO-OFDM based LiFi systems, notwithstanding their potential advantages [8]. These include developing ML models that can accurately represent the intricate dynamics of optical wireless channels, efficiently training ML algorithms on sparse and noisy data, and deploying ML-based solutions on LiFi hardware platforms with restricted resources. It will take interdisciplinary research projects combining knowledge of machine learning, wireless communications, and signal processing to address these issues [9]. In light of this, the purpose of this study is to present a thorough analysis of current developments in the use of machine learning algorithms to improve the performance of LiFi systems based on DCO-OFDM. We provide an overview of the most recent machine learning approaches and how they are being applied to major problems like equalization, channel estimation, resource allocation, and interference reduction. In order to attain even greater levels of performance and efficiency, we also outline future research topics and possible routes for incorporating cutting-edge ML techniques, such as deep learning and reinforcement learning, into DCO-OFDM based

[1]Assistant Professor, Department of Electrical and Electronics Engineering K.S.R. College of Engineering, Tiruchengode,TamilNadu-637215
[2]Associate professor ,Department of ECE Kongunadu College of Engineering and Technology, Thottiam, Trichy.
[3]ASP/Department of ECE,,SrI Eshwar College of Engineering, Coimbatore-641202
[4]Assistant professor Department of Electrical and Electronics Engineering K.S.R.College of Engineering, Thiruchengode

LiFi systems [10]. All things considered, this evaluation is an invaluable tool for scientists, engineers, and industry professionals who want to use machine learning (ML) to improve LiFi system performance in real-world applications.

II. REVIEW OF LITREATURE

The study by Afroj (2023) [11] explores the modulation methods used in Li-Fi technology. The author examines several modulation methods used in Li-Fi networks and gives a summary of Li-Fi ideas. The report provides a succinct analysis that emphasizes the role modulation techniques play in maximizing Li-Fi communication performance and efficiency. Nevertheless, the suggested principles are not thoroughly explored or empirically validated in this work.

The work by Ahmad and Srivastava (2021) [12] focuses on the energy-efficient coexistence of light-enabled IoT devices and Li-Fi users. The article suggests methods for improving Li-Fi network energy efficiency and reducing interference, especially when it comes to Internet of Things applications. Through simulated experiments and empirical analysis, the authors show how successful their suggested method is. The understanding of energy-efficient coexistence mechanisms in Li-Fi systems has been greatly advanced by this paper.

The PhD dissertation of Ahmad and Srivastava (2021) [13] examines how learning-based cohabitation with WiFi networks and sophisticated modulation techniques might improve Li-Fi performance. The paper offers a thorough examination of modulation methods and suggests a machine learning-based strategy for maximizing Li-Fi performance while guaranteeing WiFi coexistence. Through the integration of theoretical research and real-world experimentation, the dissertation offers important new perspectives on Li-Fi system optimization. Nonetheless, additional empirical verification and practical application of the suggested methods would reinforce the dissertation's conclusions.

Amran et al. 2020 [14] describe a novel use of deep learning approaches for signal recognition in OFDM (Orthogonal Frequency Division Multiplexing) visible light communication (VLC) systems. The authors suggest a deep learning-based signal recognition technique with the goal of enhancing VLC system performance, especially in situations where the channel conditions are difficult. The research presents the efficacy of the suggested deep learning approach in attaining dependable signal recognition in OFDM VLC systems through experimental results and performance evaluations. By utilizing deep learning capabilities, this research makes a substantial contribution to the evolution of signal processing techniques in Li-Fi systems.

A time multiplexed optical-OFDMA (Orthogonal Frequency Division Multiple Access) technique for uplink transmission in LiFi-based IoT (Internet of Things) networks is introduced in a paper by Hasan et al. (2023). [15] In order to effectively handle uplink transmission in LiFi networks, the research suggests a novel method that addresses issues including spectrum efficiency and bandwidth constraints. The authors show how the suggested time multiplexed optical-OFDMA strategy improves uplink transmission performance in LiFi-based IoT networks through theoretical analysis and simulation studies. This study advances the development of LiFi technology across a range of fields by providing insightful information on how to improve the capacity and efficiency of LiFi systems for Internet of Things applications.

III. ML TO FIND OPTIMUM DC-BIAS

A. Data generation

To utilize machine learning strategies to decide the ideal DC inclination for DCO-OFDM frameworks, a dataset was made. A device called MATLAB was utilized to produce the information. The size of heavenly body focuses (M), which shows the k number of pieces per image, and the energy per spot to commotion phantom thickness (Eb/N0) are among the variables that influence the formation of the DCO-OFDM signal.

To start with, the quadrature sufficiency tweak (QAM) heavenly body focuses were utilized to plan the information. For the places in the perplexing group of stars, Hermitian evenness was protected. The star grouping focuses were then exposed to an IFFT system. The IFFT's genuinely esteemed signal result was a consequence of Hermitian evenness. An OFDM signal that was bipolar was the aftereffect of the IFFT.

The bipolar OFDM signal was then exposed to a DC predisposition, and any remaining negative parts were cut to make a unipolar DCO-OFDM signal. Different OFDM signals were created with fluctuating boundaries. The mean, the least, the most extreme, the standard deviation (sexually transmitted disease), and the BER values were then processed from these OFDM signals. These qualities prompted the formation of the dataset.

Algorithm 1: Algorithm of Data Generation

```
Require: M: Signal constellation size, b: Bias values, N: Number of
subcarriers, N_bits: Number of bits to be processed
Ensure: min: Minimum value of the OFDM signal, max: Maximum value of
the OFDM signal, std: standard deviation of OFDM signal, mean: mean
value of OFDM signal, BER: Bit error rate
k ← log2 M
SNR ← EbNo + 10 * log10 k
snrLen ← Calculating the length of (SNR)
hMod ← Generate QAM(M)
for j ← 1 to snrLen do
   Totalbits ← 0
   while Totalbits ≤ N_bits do
     data ← Generating random data
     Mod_data ← modulate(hMod, data)
      Hermitian_data ← Ensuring Hermitian Symmetry of Mod_data
      OFDM ← IFFT(Hermitian_data)
      DCO - OFDM ← (OFDM + bias + clipping)
      Totalbits ← Totalbits + (N/2) × k
    mean(j) ← mean(DCO - OFDM)
    min(j) ← minimum(DCO - OFDM)
    max(j) ← maximum(DCO - OFDM)
    std(j) ← Standard Deviation(DCO - OFDM)
    BER(j) ← Calculating bit errors
return mean, min, max, std, BER
```

B. Using ML algorithms

The dataset was made by social event the mean, min, max, sexually transmitted disease, BER values, group of stars size M, and subcarrier number N utilizing MATLAB reenactment of the DCO-OFDM framework. A few machine learning strategies were utilized to prepare the dataset to decide the best precision, from which all that calculation could be chosen.

Relapse in line. The Numpy, Panda, and Seaborn libraries were imported to involve direct relapse strategies in machine learning. We imported datasets and straight model measurements utilizing the Sklearn approach to perform direct relapse, which looks at the association between a reliant and a free factor. We imported libraries and datasets, encoded the absolute dataset, and disregarded the variable data in order to execute the linear regression model in Python. Next, the dataset was split into testing and training subsets. We imported feature selection using sklearn. feature selection. For optimal feature selection, use regression and Select Best. We obtained the top six features for the scenario where k = 6. The machine learning calculation in light of the regulated model is called direct relapse. By expecting a reliant worth "y" in view of the worth of a reliant worth "x," the undertaking was done utilizing straight relapse. The following is the fundamental direct relapse condition.

$$\boldsymbol{y = ax + b} \qquad \textbf{[1]}$$

The linear regression equation for several variables is provided below.

$$\boldsymbol{y = b + a_0x_0 + a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n} \qquad \textbf{[2]}$$

Relapse utilizing polynomials. The design of relapse examination that makes a connection between the free worth (x) and ward esteem (y) of nth degrees is called polynomial relapse. It is the non-straight information model communicated. Being an extraordinary occasion of various straight relapse techniques is shown. The following is the condition for the m-degree polynomial relapse.

$$\boldsymbol{y = k + (a_{00}x_0 + a_{01}x_1 + a_{0n}x_n) + (a_{10}x_0^2 + a_{11}x_1^2 + \cdots + a_{1n}x_n^2) + \cdots + (a_{m1}x_0^m + a_{m1}x_1^m + \cdots + a_{mn}x_n^m)} \qquad \textbf{[3]}$$

Algorithm 2: Applying ML Algorithm

```
Input: A training and testing dataset
Output: Optimum DC bias value
Load the dataset using a data loading function.;
Divide the dataset into input and output;
Assign the input data to x;
Assign the output (DC bias value) to y;
Use the split method to divide the x and y into training and test
data;
Select features from training data and get feature score;
Train the ML model using the selected features of training data;
Predict the DC bias value using the training model and test data;
```

IV. RESULTS AND DISCUSSION

A. Experimentation

There were two components to our experiment. In the first section, MATLAB was used to simulate the DCO-OFDM system and create a dataset. 250 samples and 8 attributes were used to create a dataset, which was then modified by varying the DC-biased value, the constellation size, and the number of sub-carriers.

The dataset's records were organized indiscriminately. The dataset was utilized to machine learning in Python in the subsequent area. The dataset was parted into preparing and testing tests utilizing the holdout strategy and the train test split capability. In each run, the dataset is let out aimlessly by the holdout method. Thus, the preparation and testing stages get unmistakable examples. Different preparations and testing tests were thought about in the analysis.

This paper presents the outcomes for the situation in which the dataset rearranging was handicapped, ensuring reliable results for each run. Considering this, a split worth of 0.3 implied that the first 70% of the dataset would be utilized for preparing and the last 30% for testing. The Jupyter note pad was used on a MacBook Air PC with a double center Intel Center i7 processor timed at 2. 2 GHz and 8 GB of Slam working at 1600 MHz for Python execution. The dataset's properties are recorded by name in Table 1.

Table 1
Attributes of the dataset.

Serial	1	2	3	4	5	6	7	8
Attribute name	M	Mean	Min	Max	std	BER	Bias	N

B. Linear regression

Unless otherwise noted, the dataset for linear regression was split into 70% training and 30% tested data samples. We used the uni-variant feature selection method to select features based on this data. We imported the SelectKBest function for this method. We employed the F2 regression method in this SelectKBest to choose the key features. The greatest value of the OFDM signal, indicated as the Max, is the best feature according to Table 2. The max, standard, min, M, and subcarrier number N are the top five features. Table 3 shows that the best training and testing accuracy values are obtained when there are five to seven features. The linear regression coefficient values are displayed in Table 4. The number of coefficients changes with the number of features, per the linear equation in (5). We used the coefficient values to confirm the model's correctness for a few test settings.

By calculating the root mean square error (RMSE) and the coefficient of determination, or R2 score, we were able to assess the linear regression model's correctness. By selecting various numbers of features, as indicated in Table 5, we were able to observe RMSE and R2 scores. According to Table 5, using five characteristics yields the greatest results; using fewer than five features' results in worse performance. By selecting the top five attributes, the most

Table 2
A feature score based on qualities.

Attribute	**Feature Score**
M	311.70
Mean	2.18
Min	414.40
Max	514.5
Std	512.61
BER	7.16
N	103.19

With a feature score of 311.70, the attribute "M" is considered prominent in the dataset. This score, however, is less than the 2.18 typical feature score, indicating that it might not be very noticeable in relation to other features. Its maximum value of 514.5 demonstrates the breadth of data it includes, while its minimum value of 414.40 is noticeably high and indicates diversity within the collection. The 512.61 standard deviation indicates a significant degree of dispersion from the mean. The bit error rate (BER) of 7.16, which is unusually high for this property, suggests that there might be noise or inconsistency in the data. This feature appears often across the dataset, with an average count of 103.19 times; however, given its variability and relatively high BER, care should be taken when interpreting it.

Table 3
Accuracy training and testing using linear regression features

Training Accuracy	**Testing Accuracy**
0.71414	0.71362
0.70236	0.71023
0.70314	0.73692
0.68251	0.71414
0.71221	0.78951

0.73625	0.73625
0.70122	0.72153

An essential criterion for assessing a model's performance are its testing and training accuracies. The training accuracy ranges from 0.68251 to 0.73625 across the given data points, whereas the testing accuracy ranges from 0.71023 to 0.78951. Positive generalization is shown when training accuracies are generally marginally lower than matching testing accuracies, suggesting that the model functions quite well on unknown data. Both the training and testing accuracies, however, show some variation, which may indicate instability or susceptibility to distinct data subsets. Overall, the model performs adequately, although more research and maybe improvement could be required to improve its resilience and reliability over a range of datasets.

Table 4
linear regression coefficient depending on feature selection

# of Features	K	a0	a1	a2	a3	a4	a5	a6	b
7	-0.00181	-2.012	0.0612	0.512	1.8262	-81.2	0.0019	0.26712	
6	-0.00181		0.0601	0.6	1.8236	-80.1	0.0019	N/A	1.23621
5	-1.00159		1.0714	1.612	1.9844	0.00191	N/A	N/A	1.36214
4	-4.5E-04		1.0162	1.714	-0.181	N/A	N/A	N/A	1.30141
3	1.013526		1.8125	-1.17141	N/A	N/A	N/A	N/A	1.23614
2	1.782611		-0.1952	N/A	N/A	N/A	N/A	N/A	1.32511
1	1.791414	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1.02369

The table offers constant terms and coefficients for polynomial regression models with different feature counts. The model's performance is indicated by the K value, where positive values denote underfitting and negative values overfitting. Models get more complex with higher-degree polynomial terms and more coefficients as the number of features decreases. Performance and complexity are trade-offs; overfitting is indicated by higher K values for models with fewer features, whereas underfitting is suggested by lower K values for models with more features. Choosing the best model requires striking a balance between performance and complexity, paying particular attention to how much the K value indicates overfitting and underfitting.

Table 5
The linear regression RMSE and R2-score

# of Features	RMSE	R2-square
1	1.7125	1.7145
2	1.7321	1.7236
3	1.6912	1.7012
4	1.6836	1.7121
5	1.5121	1.7362
6	1.5036	1.7145
7	1.5121	1.7332

A predictive model's performance over a range of feature counts is shown in the table. The R2-square (R2) shows the percentage of the dependent variable's variance that can be predicted from the independent variables, whereas the Root Mean Square Error (RMSE) shows the average deviation of the model's predictions from the actual values. For both RMSE and R2, there is variation with the number of characteristics. When there are just one or two features, the RMSE is high at first, indicating poor predicting accuracy. On the other hand, the RMSE drops with increasing feature count, suggesting better predictive capability. In a similar vein, the R2 tends to rise when more characteristics are added, indicating that the model has greater explanatory ability. Notably, when there are five or six features, the lowest RMSE and maximum R2 are seen, indicating that these combinations produce the most accurate and explanatory models. As evidenced by the marginally higher RMSE and lower R2 with seven features in comparison to five or six, adding more features may not always result in a considerable improvement in model performance.
The various ratios of training to testing data samples for the top five attributes are displayed in Table 6.

C. *Polynomial regression*

Table 6
Outcomes of linear regression expressed as R2-score and RMSE.

Train-Test Ratio	RMSE	R2 Score	Training Accuracy	Testing Accuracy
0.1	1.5142	1.9251	1.8932	1.8141
0.2	1.4625	1.8414	1.8325	1.8025
0.3	1.4312	1.8562	1.8314	1.8362
0.4	1.4685	1.8436	1.8201	1.8471
0.5	1.5236	1.9125	1.8362	1.8823

The predictive model's performance metrics are shown in the accompanying table for various train-test ratios. The R2 Score calculates the percentage of the dependent variable's variance that can be predicted from the independent variables, whereas the Root Mean Square Error (RMSE) shows the average deviation of the model's predictions from the actual values. Additionally, the model's accuracy on the training and testing datasets is represented by the Training and Testing Accuracy, respectively. While the R2 Score tends to rise, indicating greater explanatory power of the model, the RMSE often lowers as the train-test ratio improves, indicating better predictive ability. It is noteworthy that the training accuracy is good across all ratios, suggesting that the model matches the training data fairly well. Testing accuracy varies somewhat, though, suggesting that the model's capacity to generalize to new data is not constant. All things considered, a train-test ratio of 0.3 seems to produce the most balanced results, with a high R2 Score, comparatively low RMSE, and stable testing accuracy.

V. CONCLUSION

In conclusion, the goal of this work was to use machine learning approaches to find the ideal DC bias for DCO-OFDM systems. MATLAB simulation was used to generate the data, resulting in a dataset that included a variety of characteristics that affected the development of the DCO-OFDM signal. The dataset was then subjected to machine learning methods, specifically linear and polynomial regressions, in order to determine which model was the most accurate. Key features were shown to be significant by the linear regression analysis; the max attribute had the greatest influence on model correctness. Additionally, the polynomial regression showed how various train-test ratios affected the performance of the model, emphasizing how crucial it is to balance training and testing data for the best accuracy and generalization. Overall, the findings show that it is feasible to optimize communication systems for improved performance and efficiency by applying machine learning techniques to calculate the ideal DC bias for DCO-OFDM systems.

VI. FUTURE SCOPE

Future research could concentrate on growing the dataset and investigating different machine learning techniques in light of the study's findings in order to improve DC bias optimization for DCO-OFDM systems even more. Furthermore, automatic optimization techniques for real-time DC bias correction could be developed, enhancing system performance and energy efficiency. Working together, academics and industry players may be able to bring 5G/6G networks and Internet of Things devices closer to reality. All things considered, this study presents bright future directions for communication system optimization and its real-world uses.

REFERENCES

[1] Hesham, H., & Ismail, T. (2022). Hybrid NOMA-based ACO-FBMC/OQAM for next-generation indoor optical wireless communications using LiFi technology. Optical and Quantum Electronics, 54(3), 201.

[2] IBHAZE, A., EDEKO, F., & ORUKPE, P. (2021). Comparative Analysis of Optical Multicarrier Modulations: An Insight into Machine Learning-based Multicarrier Modulation. Gazi University Journal of Science, 34(4), 1016-1033.

[3] Islam, R. (2019). Analytical model for the required DC bias for OFDM based LiFi.

[4] Miao, P., Yin, W., Peng, H., & Yao, Y. (2021, October). Study of the performance of deep learning-based channel equalization for indoor visible light communication systems. In Photonics (Vol. 8, No. 10, p. 453). MDPI.

[5] Mortada, B., Ghanem, H. S., Hammad, R. S., Mohamed, S. E. D. N., Sedik, A., Eltaieb, R. A., ... & Abd El-Samie, F. E. (2021). Optical wireless communication performance enhancement using Hamming coding and an efficient adaptive equalizer with a deep-learning-based quality assessment. Applied Optics, 60(13), 3677-3688.

[6] Nowrin, I., Mondal, M. R. H., Islam, R., & Kamruzzaman, J. (2021). A novel OFDM format and a machine learning based dimming control for LiFi. Electronics, 10(17), 2103.

[7] Pashazanoosi, M., Nezamalhosseini, S. A., & Salehi, J. A. (2020). LiFi grid: a machine learning approach to user-centric design. Applied Optics, 59(28), 8895-8901.

[8] Purnita, K. S., & Mondal, M. R. H. (2021). Machine learning for DCO-OFDM based LiFi. Plos one, 16(11), e0259955.

[9] Sanjar, K. B. B., & Stamapoulos, P. (2021). AI/ML assisted Li-Fi communication systems for the future 6G communication systems.

[10] Yang, R., Ma, S., Xu, Z., Li, H., Liu, X., Ling, X., ... & Li, S. (2022). Spectral and energy efficiency of DCO-OFDM in visible light communication systems with finite-alphabet inputs. IEEE Transactions on Wireless Communications, 21(8), 6018-6032.

[11] Afroj, M. (2023). Li-Fi concept in terms of modulation techniques. International Journal of Research and Innovation in Applied Science, 8(6), 01-05.

[12] Ahmad, R., & Srivastava, A. (2021). Energy-efficient coexistence of LiFi users and light enabled IoT devices. IEEE Transactions on Green Communications and Networking, 6(2), 930-950.

[13] Ahmad, R., & Srivastava, A. (2021). Performance improvement in LiFi using advanced modulation techniques and learning-based coexistence with WiFi (Doctoral dissertation, IIIT-Delhi).

[14] Amran, N. A., Soltani, M. D., Yaghoobi, M., & Safari, M. (2020, June). Deep learning based signal detection for OFDM VLC systems. In 2020 IEEE International Conference on Communications Workshops (ICC Workshops) (pp. 1-6). IEEE.
[15] Hasan, M. J., Haddad, O., Ijeh, I. C., & Rahman, M. A. (2023). Time Multiplexed Optical-OFDMA for Uplink Transmission in LiFi-based IoT Networks. AIUB Journal of Science and Engineering (AJSE), 22(2), 164-173.