A Novel Machine Learning Algorithm With Mathematical Modeling for Channel Estimation in VLC Systems

Sara H. ElFar, *Member, IEEE*, Maysa Yaseen, *Member, IEEE*, and Salama Ikki, *Senior Member, IEEE*

***Abstract*—In visible light communications (VLC), signal-dependent shot noise (SDSN) significantly impacts the accuracy of channel estimation. This letter introduces a simple yet novel machine learning-based estimator to overcome this challenge. A mathematical framework for the proposed estimator is developed, enabling the computation of its mean square error (MSE). Results show that, in the presence of SDSN, the proposed estimator outperforms the traditional least squares estimator (LS). In the ideal scenario without SDSN, the proposed algorithm remains slightly more efficient and accurate.**

***Index Terms*—Machine learning, visible light communication, channel estimation, signal-dependent shot noise, least squares, mean square error.**

I. Introduction

In VISIBLE light communication (VLC), accurate channel estimation is essential for maintaining reliable data transmission. However, signal-dependent shot noise (SDSN) introduces significant challenges by degrading the accuracy of conventional estimators [1], [2]. SDSN arises from the quantum nature of light, where variations in photon arrivals at the receiver cause fluctuations in the amplitude of the received signal. These fluctuations lead to inaccuracies in channel estimation, reducing the signal-to-noise ratio (SNR) and increasing the bit error rate (BER), ultimately compromising VLC system performance.

Recent studies have explored machine learning for improving VLC channel estimation. Deep learning has shown effectiveness in enhancing accuracy and robustness. For instance, [3] proposed a deep neural network (DNN)-based approach using a complex-valued neural network (CVNN) for multiple-input multiple-output (MIMO) DC-biased optical orthogonal frequency division multiplexing (DCO-OFDM) VLC systems. Their method employs a CVNN, which is specifically designed to process complex numbers directly. Similarly, the study in [4] introduced a flexible denoising convolutional neural network (FFDNet) for massive MIMO (m-MIMO) VLC channel estimation, while another work [5] proposed a deep residual convolutional denoising network (ResCBDNet) to enhance estimation accuracy in indoor m-MIMO VLC systems. ResCBDNet, trained with asymmetric and reconstruction loss functions, exhibits strong generalization capabilities and effectively handles complex noise environments.

Received 6 February 2025; revised 22 March 2025; accepted 12 April 2025. Date of publication 21 April 2025; date of current version 11 July 2025. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through its Alliance Program (NSERC Alliance). The associate editor coordinating the review of this article and approving it for publication was M. Chen. *(Corresponding author: Sara H. ElFar.)*

The authors are with the Electrical and Computer Engineering Department, Lakehead University, Thunder Bay, ON P7B 5E1, Canada (e-mail: shassan5@lakeheadu.ca; myaseen@lakeheadu.ca; sikki@lakeheadu.ca).

Digital Object Identifier 10.1109/LWC.2025.3563156

However, despite the advancements in machine learning-based VLC channel estimation, existing approaches do not consider the impact of SDSN or explore less complex machine learning models. The dependence on highly complex neural network (NN) architectures results in significant computational overhead, limiting their practicality for real-time deployment in resource-constrained environments.

Traditional least squares (LS) estimation is a computationally efficient method for VLC channel estimation that requires no prior channel knowledge [6], [7], [8], but its performance degrades under SDSN. In this letter, we enhance LS accuracy by integrating machine learning techniques, providing a more robust and efficient solution for VLC systems. To the author's best knowledge, no work in the literature has considered using machine learning to enhance the traditional LS estimation method to be used in the indoor VLC channel with the existence of the SDSN. Furthermore, no study has presented a mathematical framework for a machine learning network to estimate the channel in a VLC system, with or without considering SDSN.

Contributions: This letter makes significant advancements by moving beyond the traditional "closed-box" perspective of neural networks and establishing a clear mathematical framework. In particular, our key contributions are summarized as follows:

1) *Machine Learning-Based Estimation and SDSN Mitigation:* We introduce a lightweight machine learning-based estimator using shallow neural networks to enhance LS and maximum likelihood estimators (MLE). Our approach effectively mitigates SDSN, improving estimation accuracy while providing insights into its impact on mean square error (MSE) performance.
2) *Mathematical Framework:* We derive a polynomial representation of the trained neural network, offering a deeper analytical understanding of its behavior. Additionally, we provide a closed-form MSE expression, enabling theoretical benchmarking without repeated simulations.
3) *Efficiency and Performance Comparison:* Our model eliminates retraining needs by leveraging its mathematical formulation for direct estimation, improving computational efficiency. It also achieves competitive performance with complex models like multi-layer perception (MLP) while outperforming traditional methods in SDSN-affected scenarios.

II. System and Channel Models

This letter presents a SISO-VLC downlink transmission model with an LED transmitter and a user within a circular coverage area, focusing on the line-of-sight (LoS) link between the transmitter and photodetector (PD) [9]. The model assumes static user positioning and includes both downlink and uplink

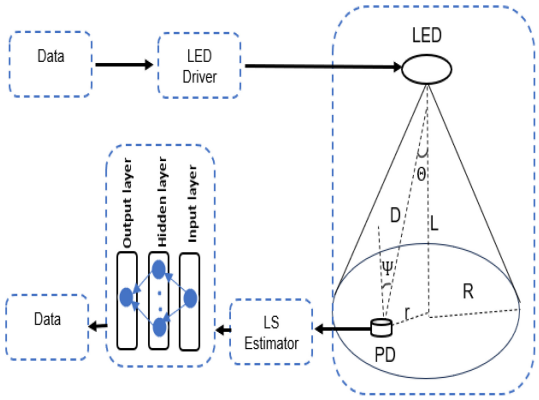

Fig. 1. The SISO VLC system and channel models.

transmission. In indoor environments, the LoS signal typically has a 7 dB advantage over the strongest Non-line-of-sight (NLoS) component [2], [10] and references therein.

The signal-dependent noise, SDSN, typically following a Poisson distribution, can be approximated by a Gaussian distribution under certain conditions, based on the central limit theorem. In the system model, the received signal at the PD is given by:

$$y = hx + \sqrt{hx}\, n_{\text{ds}} + n, \tag{1}$$

where x is the transmitted (pilot) signal, $n \sim \mathcal{N}(0, \sigma_n^2)$ is the thermal noise, and $\sqrt{hx}\, n_{\text{ds}}$ represents the SDSN with $n_{\text{ds}} \sim \mathcal{N}(0, \sigma_{ds}^2)$. SDSN is modeled as $n_{\text{ds}} \sim \mathcal{N}(0, \sigma_n^2 \zeta^2)$, where $\zeta^2 = \frac{\sigma_{ds}^2}{\sigma_n^2}$ is the shot noise scaling factor, typically ranging from 1 to 10. The power of SDSN is proportional to both the input power x and the channel gain h, unlike thermal noise, which is independent of these factors [11], [12].

The channel gain h, representing the transmission link between the LED transmitter and the PD receiver, is modeled as described in [13], [14] and references therein:

$$h = \frac{A_{pd}\eta(m+1)}{2\pi D^2} \cos^m(\Theta)\, T_s(\Psi) g(\Psi) \cos(\Psi), \tag{2}$$

where Ψ denotes the angle of incidence relative to the surface normal of the PD, and Θ is the angle of irradiance with respect to the transmitter's surface normal. In this equation, $T_s(\Psi)$ represents the gain of the receiver's optical filter, and $g(\Psi) = \frac{n^2}{\sin^2(\Phi_{\text{FOV}})}$ is the gain of the optical concentrator. The gain $g(\Psi)$ depends on the refractive index n of the concentrator and the field of view (FOV) angle Φ_{FOV}. Note that $g(\theta_{\text{rx}}) = 0$ when $\theta_{\text{rx}} > \Phi_{\text{FOV}}$. Additionally, it A_{pd} represents the detection area of the PD, and η denotes the average responsivity of the receiver. The Euclidean distance between the transmitter and the receiver, D, is $D = \sqrt{L^2 + r^2}$, where L is the vertical distance from the LED to the PD surface and r is the horizontal distance from the center of the LED cell to the PD, as illustrated in Fig. 1. Finally, the Lambertian emission order m is expressed as $m = \frac{-1}{\log_2(\cos(\Phi_{1/2}))}$, where $\Phi_{1/2}$ denotes the semi-angle of the LED's emission.

III. Channel Estimation

Traditional RF-based communication methods, such as LS, MLE, maximum a posteriori (MAP), minimum mean square error (MMSE), and least minimum mean square error (LMMSE) estimators, can also be applied to VLC systems. However, VLC channels possess unique characteristics, such as positive and real signal requirements and the impact of SDSN [5], [15], which can limit the effectiveness of these techniques. Advanced methods like MAP and LMMSE require prior statistical information about the channel, which is often unavailable in practical VLC systems, particularly in large m-MIMO scenarios. Additionally, optimal methods such as MMSE rely on long pilot sequences, further increasing computational complexity and reducing practicality [5], [16].

This letter addresses channel estimation in VLC systems by integrating an NN with the conventional LS estimator. The neural network mitigates SDSN-related performance degradation in LS estimation. After training, polynomial regression approximates the NN output, enabling a closed-form MSE calculation. This provides an efficient and accurate solution for VLC channel estimation.

We assume that the number of pilot symbols is denoted by N, and the LED transmits a pilot vector $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$, where $[.]^T$ represents the transpose operation. As a result, the received signal vector $\mathbf{y} = [y_1, y_2, \ldots, y_N]^T$ can be expressed as:

$$\mathbf{y} = h\mathbf{x} + \sqrt{h\, \text{diag}(\mathbf{x})}\mathbf{n}_{ds} + \mathbf{n}, \tag{3}$$

where $\text{diag}(\mathbf{x})$ is a $N \times N$ diagonal matrix containing the elements of the transmitted signal vector $\mathbf{x}$ along the main diagonal. The vectors $\mathbf{n} = [n_1, n_2, \ldots, n_N]^T$, $\mathbf{n}_{\text{ds}} = [n_{\text{ds1}}, n_{\text{ds2}}, \ldots, n_{\text{dsN}}]^T$ represent independent noise components. The entries of $\mathbf{n}$ are independent and identically distributed (i.i.d.) random variables, following $\mathbf{n} \sim \mathcal{N}(0, \sigma_n^2 I_N)$,where I_N is the identity matrix of size N. Similarly, the entries of $\mathbf{n}_{ds}$ follow $\mathbf{n}_{ds} \sim \mathcal{N}(0, \zeta^2 \sigma_n^2 I_N)$, where ζ^2 represents the scaling factor for the SDSN. Finally, an important characteristic of VLC systems is that the transmitted signal must be both positive and real. Therefore, all pilot symbols must satisfy the condition $x_i > 0, \forall i \in \{1, 2, \ldots, N\}$.

A. Least Squares Estimator

LS is a low-complexity estimator that does not need prior information. The estimated $\hat{h}$ using the LS estimator can be expressed as:

$$\hat{h}_{\text{LS}} = \frac{\mathbf{x}^T \mathbf{y}}{\|\mathbf{x}\|^2}. \tag{4}$$

To derive the MSE of the LS estimator, we first substitute (3) into (4). This allows us to express the LS estimator of the channel gain, h, as:

$$\hat{h}_{\text{LS}} = \frac{\mathbf{x}^T \mathbf{y}}{\|\mathbf{x}\|^2} = h + \frac{\mathbf{x}^T}{\|\mathbf{x}\|^2} \sqrt{h\, \text{diag}(\mathbf{x})}\mathbf{n}_{\text{ds}} + \frac{\mathbf{x}^T}{\|\mathbf{x}\|^2}\mathbf{n}. \tag{5}$$

The error of the LS estimator can be expressed as:

$$\epsilon_{\text{LS}} = \hat{h}_{\text{LS}} - h = \frac{\mathbf{x}^T}{\|\mathbf{x}\|^2} \left(\sqrt{h\, \text{diag}(\mathbf{x})}\mathbf{n}_{\text{ds}} + \mathbf{n} \right). \tag{6}$$

From (6), we can deduce that the error is a Gaussian random variable with zero mean and variance $\sigma_{\epsilon_{\text{LS}}}^2$, i.e., $\epsilon_{\text{LS}} \sim \mathcal{N}(0, \sigma_{\epsilon_{\text{LS}}}^2)$. Therefore, the MSE of the LS estimator is simply the variance of the error, which can be expressed as:

$$\mathrm{MSE}_{\mathrm{LS}} = \sigma^2_{\epsilon_{\mathrm{LS}}} = \frac{\sigma_n^2 \sum_{i=1}^{N} x_i^2 \left(1 + \zeta^2 h x_i\right)}{\left(\sum_{i=1}^{N} x_i^2\right)^2}. \tag{7}$$

Assuming, without loss of generality, that the transmitted pilots $x_i = p$ for all i, the MSE can then be written as [2]:

$$\mathrm{MSE}_{\mathrm{LS}} = \sigma^2_{\epsilon_{LS}} = \frac{\sigma_n^2 \left(1 + \zeta^2 h p\right)}{N p^2}. \tag{8}$$

It is worth mentioning that in the absence of the SDSN, i.e., $\zeta^2 = 0$, the MSE is inversely proportional to p^2 while, in the presence of the SDSN, the MSE is inversely proportional to p, indicating the harmful effect of the SDSN. Furthermore, increasing the number of pilots, N, will improve the estimation performance, and the presence or absence of the SDSN has no relation impact on the number of pilots.

B. Integrated Machine Learning and Least Squares Estimator

The proposed method begins by estimating the channel gain through a straightforward traditional estimator, such as the LS estimator, or another common approach like the MLE. In this letter, the LS estimator is applied initially. The output from the LS estimation is then refined and improved by passing it through an NN.

Consider a feed-forward NN consisting of a single hidden layer with N_h neurons and a single output layer with one neuron. The output layer uses a linear activation function, while the hidden layer utilizes a nonlinear activation function. The hidden layer's output for each training sample is computed as:

$$\boldsymbol{y}_h = \Lambda^{(1)}\Big(\boldsymbol{W}\hat{h}_{\mathrm{LS}} + \boldsymbol{b}\Big). \tag{9}$$

where $\Lambda^{(1)}$ represents the hidden layer's activation function, $\boldsymbol{W} = [w_1, \ldots, w_k, \ldots w_{N_h}]^T$ is the weight matrix of dimensions $N_h \times 1$, and $\boldsymbol{b} = [b_1, \ldots, b_k, \ldots b_{N_h}]^T$ is the bias vector of size $N_h \times 1$. Next, the output layer generates the final output for each input as $\hat{h}_{\mathrm{NN}} = \Lambda^{(2)}(\boldsymbol{V}\boldsymbol{y}_h + \beta)$, where $\Lambda^{(2)}$ serves as the activation function of the output layer, $\boldsymbol{V} = [v_1, \ldots, v_k, \ldots v_{N_h}]$ is the weight matrix of dimensions $1 \times N_h$, and β is the scalar bias for the output layer.

In the training process of the ANN model, the set of parameters $\Gamma \triangleq \{\boldsymbol{W}, \boldsymbol{V}, \boldsymbol{b}, \beta\}$ is adapted to achieve the optimum Γ_{opt} that minimizes the cost function, which is the MSE between the output of the ANN and the real value of the channel gain; i.e., h therefore, the optimum set of parameters of the ANN model can be optimized as:

$$\Gamma_{\mathrm{opt}} = \min_{\Gamma} \mathbb{E}\left\{\left(h - \hat{h}_{\mathrm{NN}}\right)^2\right\}. \tag{10}$$

It is worth mentioning that the model's weights and biases are adjusted iteratively with each training sample. Considering M training samples, the optimal parameter Γ_{opt} is averaged over all samples, allowing for continuous refinement and improved accuracy.

IV. Comprehensive Mathematical Framework and MSE Analysis

In this section, we analyze the neural network by examining its layers at the neuron level to derive an equivalent polynomial form. Using a Taylor series expansion, we approximate the network's behavior and derive a closed-form expression for the theoretical MSE of the estimated channel output, $\hat{h}_{\mathrm{NN}}$, during the testing phase. This approach utilizes the fixed weights and biases from the training phase, enabling precise MSE performance assessment under the trained model parameters. As previously mentioned, we consider a feed-forward neural network with a single hidden layer containing N_h neurons and an output layer with a single neuron [17].

The output of the k^{th} neuron in the hidden layer, given that the activation function for the hidden layer is tanh, is[1]:

$$y_k = \tanh(u_k) = \tanh\Big(w_k \hat{h}_{\mathrm{LS}} + b_k\Big) = \frac{e^{u_k} - e^{-u_k}}{e^{u_k} + e^{-u_k}}, \tag{11}$$

where $u_k = w_k \hat{h}_{\mathrm{LS}} + b_k$, w_k and b_k are the weights and bias at the k^{th} neuron in the hidden layer, respectively. The output of the NN, $\hat{h}_{\mathrm{NN}}$ is a linear combination of the activations of the hidden neurons multiplied by linear weights of the output layer v_k plus the output layer bias β [18]. It can be expressed as:

$$\hat{h}_{\mathrm{NN}} = \sum_{k=1}^{N_h} v_k y_k + \beta = \sum_{k=1}^{N_h} v_k \tanh\Big(w_k \hat{h}_{\mathrm{LS}} + b_k\Big) + \beta. \tag{12}$$

To find an approximate equivalent to the NN we use the Taylor expansion series.

A. Taylor Series Approximation of $tanh(u_k)$

The $\tanh$ function can be approximated by its Taylor series expansion around $u_k = 0$. The Taylor expansion of $\tanh(u_k)$ can be obtained as:

$$\tanh(u_k) = \sum_{n=0}^{\infty} \frac{d^n \tanh(0)}{d u_k^n} \frac{u_k^n}{n!}, \tag{13}$$

where $\frac{d^n \tanh(0)}{d u_k^n}$ represents the n^{th} derivative of $\tanh(u_k)$ evaluated around $u_k = 0$ and n represents the order of Taylor expansion. Using this series for the activation function, the output of the network can be rewritten as:

$$\hat{h}_{\mathrm{NN}} = \sum_{k=1}^{N_h} v_k \sum_{n=0}^{\infty} \frac{d^n \tanh(0)}{d u_k^n} \frac{u_k^n}{n!} + \beta. \tag{14}$$

Therefore, the MSE of $\hat{h}_{\mathrm{NN}}$ at the testing phase after training is calculated as $\mathrm{MSE}_{\mathrm{NN}} = \mathbb{E}\{(h - \hat{h}_{\mathrm{NN}})^2\}$. After some mathematical manipulations, the MSE can be expressed as:

$$\begin{aligned}\mathrm{MSE}_{\mathrm{NN}} = {} & \mathbb{E}\left[h^2\right] \\ & - \mathbb{E}\left[2h\left(\sum_{k=1}^{N_h} v_k \sum_{n=0}^{\infty} \frac{d^n \tanh(0)}{d u_k^n} \frac{\left(w_k \hat{h}_{\mathrm{LS}} + b_k\right)^n}{n!} + \beta\right)\right] \\ & + \mathbb{E}\left[\left(\sum_{k=1}^{N_h} v_k \sum_{n=0}^{\infty} \frac{d^n \tanh(0)}{d u_k^n} \frac{\left(w_k \hat{h}_{\mathrm{LS}} + b_k\right)^n}{n!} + \beta\right)^2\right],\end{aligned} \tag{15}$$

where $(w_k \hat{h}_{\mathrm{LS}} + b_k)^n$ the term can be represented by a polynomial as $(w_k \hat{h}_{\mathrm{LS}} + b_k)^n = \sum_{j=0}^{n} \binom{n}{j} (w_k \hat{h}_{\mathrm{LS}})^{(n-j)} b_k^j$. After applying the expectation and performing the necessary calculations on (15), the closed-form expression for $\mathrm{MSE}_{\mathrm{NN}}$ can be calculated as in (16) at the bottom of the next page, where the binomial function is defined as $\binom{a}{s} = \frac{a!}{s!(a-s)!}$, and since $\hat{h}_{\mathrm{LS}}$ is a Gaussian random variable with mean h and

[1]The generalization to any activation function or additional hidden layers is straightforward and is omitted here due to space constraints.

Algorithm 1 Integrated Machine Learning and Least Squares Estimator

Data Preparation:

1) Generate the training data, we consider an indoor area of size $4\,\mathrm{m} \times 4\,\mathrm{m}$, and divide it into small blocks of $4\,\mathrm{cm} \times 4\,\mathrm{cm}$. We then calculate the true channel for each block, assuming PD in this block. After adding noise, we estimate $\hat{h}_{\mathrm{LS}}$ for each block from the received signal. This forms the training dataset, where $\hat{h}_{\mathrm{LS}}$ is the input, and the true channel h serves as the output.
2) Normalize the generated data and divide it into training (80%) and testing (20%) (The proposed algorithm can be directly applied to similar datasets.)

Neural Network Training and Testing:

1) Initialize all NN state weights and biases.
2) Use $\hat{h}_{\mathrm{LS}}$ as input to the NN with the true channel values h as targets from the generated dataset.
3) Train the NN with stochastic gradient descent (SGD) to minimize the MSE between h and $\hat{h}_{\mathrm{LNN}}$, using the cost function in (10) averaged over all training samples.
4) Update the NN's weights and biases to minimize the MSE.
5) Save the trained model with optimized weights and biases for testing.

TABLE I
THE PARAMETERS OF THE NN

Parameters	Values
Size of the training data	8000
Size of the validation data	2000
Number of epochs	100
Number of hidden layers	1
Learning rate	0.001
Number of hidden layer neurons (N_h)	10
Loss function	MSE
Activation function	tanh

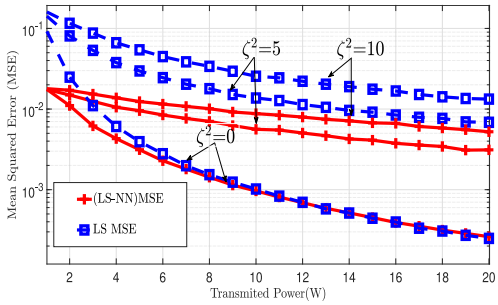

Fig. 2. Comparison of the LS estimator and the integrated LS-NN estimator at different values of ζ^2.

variance $\sigma^2_{\epsilon_{\mathrm{LS}}}$, i.e., $\hat{h}_{\mathrm{LS}} \sim \mathcal{N}(h, \sigma^2_{\epsilon_{\mathrm{LS}}})$, its n^{th} moment, i.e., $\mathbb{E}[(\hat{h}_{\mathrm{LS}})^n]$, can be found in many references, e.g., [19].

It is worth mentioning that, as we will see in Section V, it is sufficient to use only the first two terms of the infinite series represented in (16). In fact, the first two terms will provide an accuracy of more than 99%.

A second remark is that we can see from equations (16) that the final MSE depends on the value of h as well as the weights and biases of the trained NN. Finally, to apply our proposed integrated estimation technique, the steps in Algorithm 1 should be followed. It is beneficial to note that for multiple test samples, the average $\mathrm{MSE}_{\mathrm{NN}}$ should be computed across all samples to assess overall performance. Moreover, our algorithm can be applied to enhance other estimation methods such as MLE, LMMSE, and MMSE; however, this increases complexity due to their need for more statistical information.

V. Simulation Results and Discussion

This section presents a numerical analysis of SISO-VLC systems, examining the impact of SDSN on channel estimation between the transmitter and the PD. We assess the LS estimator and its hybrid integration with an NN through both simulations and analytical evaluations. Furthermore, to generalize our proposed algorithm, we investigate how the NN enhances the performance of the MLE (which is a direct extension to the presented approach). Furthermore, we compare the effectiveness of the proposed LS-shallow NN with the LS-MLP NN. The MLP, commonly used in channel estimation, features two hidden layers or more to analyze the impact of deeper architectures.

The simulations, conducted in MATLAB, involve approximately 10^6 repeated Monte Carlo iterations per figure. The system parameters are set as follows: a field of view $\phi_{\mathrm{FOV}=70^\circ}$, transmission distance $L = 2$ m, and a responsivity factor $\eta = 0.4\mathrm{A/W}$. For the NN parameters, refer to Table I.

In Fig. 2, the average $\mathrm{MSE}_{\mathrm{NN}}$ and, $\mathrm{MSE}_{\mathrm{LS}}$ for all testing samples, the corresponding estimated channels $\hat{h}_{\mathrm{NN}}$ and $\hat{h}_{\mathrm{LS}}$, are plotted against the transmitted power at thermal noise variance $\sigma_n^2 = 0.1$. From this figure, we observe that as the SDSN scaling factor ζ^2 increases, the MSE rises across all estimation methods. However, the proposed NN-based algorithm effectively reduces the MSE, particularly at higher ζ^2 values, outperforming the LS estimator. Additionally, at lower transmitted power levels, the NN-based method shows superior performance. Even in the absence of SDSN, it slightly outperforms the LS estimator, highlighting its robustness in handling material imperfections.

Fig. 3 compares the MSE from the NN estimation using the full $\tanh$ activation function and its Taylor expansion with the theoretical mathematical expression of the MSE at $\zeta^2 = 5$. The results demonstrate that the MSE obtained using the NN agrees with the theoretical MSE from (16), making it applicable for

$$\mathrm{MSE}_{\mathrm{NN}} = h^2 - 2h\left(\sum_{k=1}^{N_h}\sum_{n=0}^{\infty}\sum_{j=0}^{n}\frac{d^n\tanh(0)}{du_k^n}\frac{v_k}{n!}\binom{n}{j}(w_k)^{(n-j)}\mathbb{E}\left[\left(\hat{h}_{\mathrm{LS}}\right)^{(n-j)}\right]b_k^j + \beta\right)$$
$$+\sum_{k=1}^{N_h}\sum_{k'=1}^{N_h}\sum_{n=0}^{\infty}\sum_{j=0}^{n}\sum_{n'=0}^{\infty}\sum_{j'=0}^{n'}\frac{d^n\tanh(0)}{du_k^n}\frac{v_k}{n!}\binom{n}{j}(w_k)^{(n-j)}b_k^j\frac{d^{n'}\tanh(0)}{du_{k'}^{n'}}\frac{v_{k'}}{n'!}\binom{n'}{j'}(w_{k'})^{(n'-j')}b_{k'}^{j'}\mathbb{E}\left[\left(\hat{h}_{\mathrm{LS}}\right)^{(n'-j')}\left(\hat{h}_{\mathrm{LS}}\right)^{(n-j)}\right]$$
$$+2\beta\sum_{k=1}^{N_h}\sum_{n=0}^{\infty}\sum_{j=0}^{n}\frac{d^n\tanh(0)}{du_k^n}\frac{v_k}{n!}\binom{n}{j}(w_k)^{(n-j)}\mathbb{E}\left[\left(\hat{h}_{\mathrm{LS}}\right)^{(n-j)}\right]b_k^j + \beta^2. \tag{16}$$

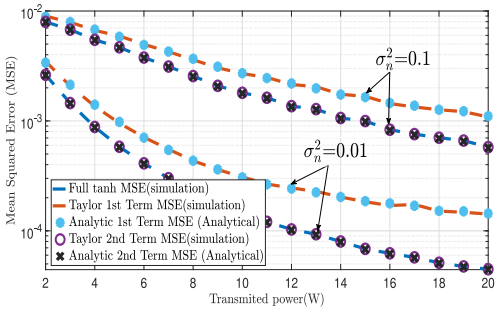

Fig. 3. Comparison of NN simulation MSE with analytical MSE across varying terms in the Taylor expansion at different values of σ_n^2.

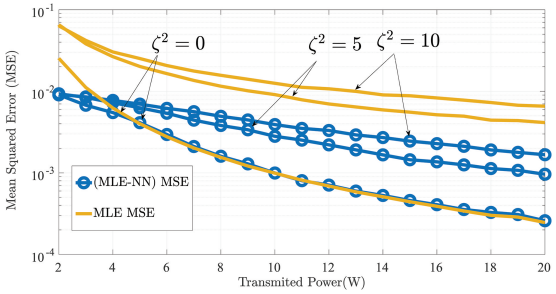

Fig. 4. Comparison of MLE and integrated MLE-NN estimator at different values of ζ^2.

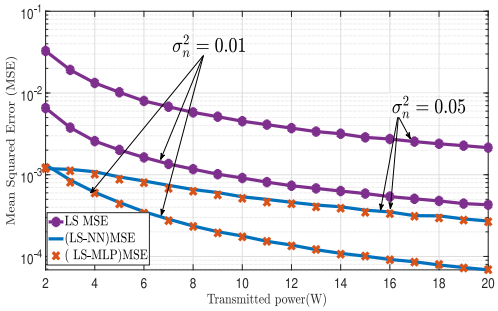

Fig. 5. Comparison of LS, LS-NN, and LS-MLP at different values of σ_n^2.

practical use. Additionally, you can see from the figure that when the σ_n^2 increases, the MSE increases in all techniques. Furthermore, the figure shows that, although the analytical expression for the MSE involves an infinite series, using only the first two terms yields highly accurate results. This trend also holds true for other activation functions.

Fig. 4 illustrates that the proposed algorithm, utilizing a shallow NN, enhances the MLE performance at different ζ^2 values when $\sigma_n^2 = 0.1$. This figure demonstrates that incorporating NN improves estimation accuracy in the presence of SDSN noise. Additionally, increasing the SDSN scaling factor leads to a higher MSE, while the NN further enhances estimation performance. The MLE estimator in the presence of SDSN is more complex to find, as described in [2]. Moreover, it requires knowledge of the likelihood function of the received signal, making it computationally demanding. In contrast, the LS estimator is less complex and mathematically tractable.

Fig. 5 presents a performance comparison of LS estimation, LS-NN, and LS-MLP across different values of σ_n^2 for $\zeta^2 = 10$. The results indicate that both LS-NN and LS-MLP consistently outperform LS alone at any σ_n^2. The MLP model comprises two hidden layers with 10 neurons each. However, the performance improvement of LS-NN and LS-MLP remains closely similar. Therefore, given the problem constraints, a single-layer NN is sufficient, avoiding unnecessary complexity in the neural network.

VI. Conclusion

In this letter, we introduce the first integrated technique that combines traditional LS estimation with an NN to estimate the channel in environments affected by SDSN, which degrades LS estimator performance. We derive the closed-form expression for the MSE of the estimated channel parameter using this integrated approach and validate it against simulation results, which closely align with our theoretical predictions. This validation enables the direct application of the $\mathrm{MSE_{NN}}$ rule once the neural network's weights and biases are determined, offering an efficient way to combine traditional estimation and machine learning for improved channel estimation.

References

[1] Y. Wang et al., "Band-limited Nyquist pulse design for visible light communication systems with signal-dependent noise," *IEEE Internet Things J.*, vol. 11, no. 16, pp. 27651–27663, Aug. 2024.
[2] M. Yaseen et al., "Visible light communication with input-dependent noise: Channel estimation, optimal receiver design and performance analysis," *J. Lightw. Technol.*, vol. 39, no. 23, pp. 7406–7416, Dec. 1, 2021.
[3] S. K. G. S. Bose and A. Kumar, "DNN based channel estimation for indoor MIMO multipath visible light communication system," in *Proc. 14th Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*, 2023, pp. 1–7.
[4] Z. Gao et al., "FFDNet-based channel estimation for massive MIMO visible light communication systems," *IEEE Wireless Commun. Lett.*, vol. 9, no. 3, pp. 340–343, Mar. 2020.
[5] M. H. Rahman et al., "Channel estimation for indoor massive MIMO visible light communication with deep residual convolutional blind denoising network," *IEEE Trans. Cogn. Commun. Netw.*, vol. 9, no. 3, pp. 683–694, Jun. 2023.
[6] J. C. Estrada-Jiménez et al., "Superimposed training-based channel estimation for MISO optical-OFDM VLC," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 6161–6166, Jun. 2019.
[7] S. H. ElFar, M. Yaseen, and S. Ikki, "Insights into visible light positioning: Range tracking and Bayesian cramér-Rao lower bound analysis," *IEEE Commun. Lett.*, vol. 28, no. 9, pp. 2056–2060, Sep. 2024.
[8] S. M. J. Pradhan et al., "Time-varying channel estimation for ACO-OFDM VLC over mobile environment," *IEEE Trans. Veh. Technol.*, vol. 73, no. 8, pp. 11556–11567, Aug. 2024.
[9] Q. Gao et al., "Joint transceiver and offset design for visible light communications with input-dependent shot noise," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 2736–2747, May 2017.
[10] L. Zeng et al., "High data rate multiple input multiple output (MIMO) optical wireless communications using white led lighting," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 9, pp. 1654–1662, Dec. 2009.
[11] H. Chen and Z. Xu, "A two-dimensional constellation design method for visible light communications with signal-dependent shot noise," *IEEE Commun. Lett.*, vol. 22, no. 9, pp. 1786–1789, Sep. 2018.
[12] S. M. Moser, "Capacity results of an optical intensity channel with input-dependent gaussian noise," *IEEE Trans Inf. Theory*, vol. 58, no. 1, pp. 207–223, Jan. 2012.
[13] L. Yin et al., "Performance evaluation of non-orthogonal multiple access in visible light communication," *IEEE Trans. Commun.*, vol. 64, no. 12, pp. 5162–5175, Dec. 2016.
[14] Z. Ghassemlooy et al., *Optical Wireless Communications: System and Channel Modeling with MATLAB*, 1st ed. Boca Raton, FL, USA: CRC Press, May 2017.
[15] L. Wu et al., "Channel estimation for optical-OFDM-based multiuser MISO visible light communication," *IEEE Photon. Technol. Lett.*, vol. 29, no. 20, pp. 1727–1730, Oct. 2017.
[16] X. Shi et al., "Adaptive least squares channel estimation for visible light communications based on tap detection," *Opt. Commun.*, vol. 467, Jul. 2020, Art. no. 125712.
[17] P. Morala et al., "Towards a mathematical framework to inform neural network modelling via polynomial regression," *Neural Netw.*, vol. 142, pp. 57–72, Oct. 2021.
[18] P. Morala et al., "NN2Poly: A polynomial representation for deep feed-forward artificial neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 1, pp. 781–795, Jan. 2025.
[19] A. Winkelbauer, "Moments and absolute moments of the normal distribution," 2012, *arXiv:1209.4340*.