

Article

Intelligent Machine Learning–Based Routing with Feature Extraction for Optical Benes Networks

Li Zhao ^{1,2,*} , Bin Hu ¹, Syed Baqar Hussain ³ , Amber Sultan ³  and Yong Kong ¹

¹ School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China; m325124106@sues.edu.cn (B.H.); kky7757@sues.edu.cn (Y.K.)

² Anhui Zhiguo Intelligent Technology Co., Ltd., Chuzhou 239061, China

³ School of Systems and Technology, University of Management and Technology, Lahore 54770, Pakistan; baqar.hussain@umt.edu.pk (S.B.H.); amber.sultan@umt.edu.pk (A.S.)

* Correspondence: zl@sues.edu.cn

Abstract

Optical Benes networks are effective switching architectures for high-capacity communication systems. However, conventional routing algorithms primarily emphasize connectivity while often overlooking path quality, which often results in severe transmission loss along worst-case paths. To address this limitation, we propose an intelligent routing framework that integrates a feature extraction module with the K-Nearest Neighbors (KNN) algorithm. The proposed method guides path selection more effectively and avoids worst-case routing scenarios through effective preprocessing and feature extraction from routing tables. A 30 Gbps PAM4 transmission system is simulated to evaluate the proposed approach. For performance comparison, conventional routing methods, as well as Support Vector Machine (SVM), and Convolutional Neural Network (CNN) routing methods are considered. The results reflect significant improvements in routing accuracy (from 55% to 72.85%) with KNN, which significantly outperforming the CNN (52.23%) and SVM (51.06%) approaches while achieving the lowest computational cost of all tested methods (0.1–1 ms per iteration). Furthermore, the proposed approach reduces the power penalties, enhances the Extinction Ratio (EXT), and lowers the Symbol Error Rate (SER). Analysis using eye diagrams confirms superior signal integrity at lower received power levels. These findings demonstrate that the feature-enhanced KNN routing algorithm is an efficient and intelligent solution that not only ensures connectivity but also optimizes path quality, paving the way for scalable, high-speed optical Benes networks.

Keywords: optical Benes networks; routing algorithm; K-Nearest Neighbors (KNN); feature extraction; machine learning in optical networks; path optimization; Symbol Error Rate (SER); Extinction Ratio (EXT); PAM4 modulation



Received: 6 May 2026

Revised: 4 June 2026

Accepted: 25 June 2026

Published: 28 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

Attribution (CC BY) license.

1. Introduction

Recently, machine learning has been applied to chip design to improve performance and reduce power consumption [1,2]. To address such challenges, a 32×32 on-chip optical Benes network has been successfully designed and implemented. Notably, the routing problem for optical Benes networks can be formulated as a classification task, making it well suited for machine learning–based solutions [3]. Despite these advances, one critical challenge remains: severe signal loss along the worst-case path, mainly caused by the limitations of conventional routing algorithms.

Although a variety of intelligent routing algorithms have been developed to circumvent worst-case routing paths and maintain signal quality above the receiver sensitivity threshold [3,4], the intrinsic computational complexity of Benes optical switches makes it highly challenging to accurately identify and isolate such worst-case paths using conventional algorithmic cost metrics [5–8]. As a result, existing routing strategies tend to focus merely on guaranteeing path connectivity rather than optimizing optical path quality; this compromise severely limits power performance and degrades signal integrity [9,10].

Feature extraction methods, when incorporated with machine learning (ML), provide a promising perspective for addressing this worst-case path problem. Traditional studies on Benes networks often rely on simplistic, one-dimensional features to determine worst-case scenarios, such as the maximum number of path crossings [11] or identifying all-crossing patterns across the $2 \log_2 N - 1$ stages of the network [12]. However, such approaches fail to capture the multi-dimensional characteristics of actual optical performance. To address this, our algorithm extracts three specific features: f_1 , f_2 and f_3 , corresponding to metrics for crosstalk, power penalty, and path dependency, respectively. Although compact, this three-dimensional feature space effectively captures the recursive construction properties of Benes networks.

Routing in optical Benes networks is fundamentally a classification problem, a domain where the K-Nearest Neighbors (KNN) algorithm has demonstrated significant efficacy. Despite its widespread success in various applications [13–15], the deployment of KNN in optical Benes networks has received limited attention. This gap is primarily due to the inherent Boolean routing tables of these networks, which exhibit insufficient data correlation and thereby limit the direct applicability of traditional KNN. To overcome this limitation, this paper proposes an enhanced KNN-based routing scheme that incorporates three highly correlated features—augmenting the conventional approach by two additional features—to optimize routing performance in optical Benes networks.

The rest of the paper is organized as follows. The fundamentals and limitations of Benes networks are covered in Section 2, and the requirements of the KNN model are described in Section 3. The proposed algorithm and details of each sub-module are presented in Section 4. The simulation setup and comparative results are shown in Section 5. A discussion and conclusion are provided in Sections 6 and 7, respectively.

2. Motivation

2.1. AI Prediction of the Worst-Case Path

Routing table categorization has become increasingly important in photonic chips in recent years. In these applications, machine learning and deep learning techniques are frequently employed for detection and classification. For such categorization tasks, hidden information in path signals can be extracted using intelligent algorithms [16–20]. The Benes path signals obtained from our simulation via Lumerical INTERCONNECT contain unique, discernible patterns of physical impairments. Here, we define physical impairment features as follows: f_1 (first Cross position) encodes early-stage crosstalk, f_2 (total Cross count) encodes cumulative power loss, and f_3 (first Bar position) encodes inter-path dependency. Instead of pursuing a rigorous combinatorial proof, our ML approach statistically maps extracted structural features to practical physical impairments, functioning as an efficient distance-based classifier to circumvent the NP-hard search space. Compared to other ML techniques, K-Nearest Neighbors (KNN) is computationally efficient, easy to train, and highly effective for distance-based classification tasks; therefore, it is adopted in this work for path categorization.

For the $N!$ possible routing tables, finding the precise routes for this problem necessitates a thorough approach, which is very time-consuming and labor-intensive. This work

simplifies the problem by defining the worst-case path as one whose EXT is lower than the average EXT value of the comprehensive dataset. In this paper, two categories of paths are considered: detectable paths and undetectable ones. The mean value employed in our definition is calculated only over detectable paths, whereas the definition of maximum loss may involve paths that are undetectable. Consequently, our proposed definition is inherently superior to the conventional maximum-loss-based one.

2.2. The Benes Network and Boolean Routing Table

As a Benes network is a recursive Clos network, a set of 2×2 switching units makes up each input and output stage, and two smaller Clos networks recursively replace the middle stage. A Clos network can continue replacing each intermediate module in this manner until the intermediate stage reaches a switch size of 2×2 [21]. Every switch element (SE) in a Benes network operates in one of two states: Cross or Bar. The Boolean routing table uses '0' to indicate the Bar state and '1' to indicate the Cross state. The raw routing matrix exhibits three fundamental characteristics [22]:

1. It has a fixed dimension of $N \times M$, where N is the number of rows and M is the number of columns;
2. The matrix strictly consists of binary values (0s and 1s), representing the Bar and Cross states, respectively;
3. Owing to the poor correlation coefficient, the Boolean routing table is a non-linear classification problem.

2.3. Feature Extraction and Representations

To accurately predict the power loss induced by crosstalk and to evaluate power efficiency and path dependency, three distinctive features, denoted as f_1 , f_2 and f_3 , are extracted from the raw routing matrix. Specifically, f_1 represents the position of the first '1' in a column, corresponding to the first occurrence of the Cross state; f_2 refers to the total number of 1s, that is, the overall number of switching elements (SEs) operating in the Cross state throughout the network; and f_3 indicates the position of the first '0' in a row, that is, the first emergence of the Bar state. The ranges of f_1 , f_2 and f_3 are $[0, 2 \log_2 N - 1]$, $[0, N \log_2 N - N/2]$, and $[0, N]$, respectively. These features are chosen according to the following physical considerations.

First-order crosstalk occurs in the optical path primarily when two signals simultaneously cross the same SE. Since an SE is in the Cross state when its value is 1, f_1 effectively determines the stage at which the signal first experiences severe crossing crosstalk. Secondly, the number of Bar states fundamentally influences the power efficiency in the optical channel. Since the total number of SEs in the network is fixed, tracking the total number of Cross states f_2 inversely yet accurately reflects the proportion of Bar states, thus serving as a reliable metric for evaluating overall power efficiency.

Finally, "path dependency" describes whether the positions of two paths overlap; if they do, the switching element is in the Bar state. In other words, a switch element's '0' state prevents crosstalk between switching elements [23]. The crossings in a Benes switch's structure are the only source of crosstalk. As a result, f_3 denotes reduced path dependency.

Consequently, by extracting these three physical impairment features to form a 3D Euclidean space, a KNN classifier can be effectively trained. This approach not only bypasses the limitations of Boolean correlation but also greatly enhances the physical interpretability of optical impairments within the ML representation. A KNN algorithm is considered adequate when its accuracy exceeds 70%; otherwise, it is considered ineffective [24].

3. Principle

The algorithm shown in Figure 1a is a traditional routing method based on the two-color bipartite matching problem. It computes the routing table matrix S for connections from inputs $I_1, I_2, \dots, I_N$ to outputs $O_1, O_2, \dots, O_N$ in polynomial time, using recursive or parallel approaches. However, this traditional routing algorithm has a drawback in that it treats all optical paths equally. As the number of ports N increases, the selection of the routing table matrix leads to significant end-to-end performance fluctuations along the optical paths.

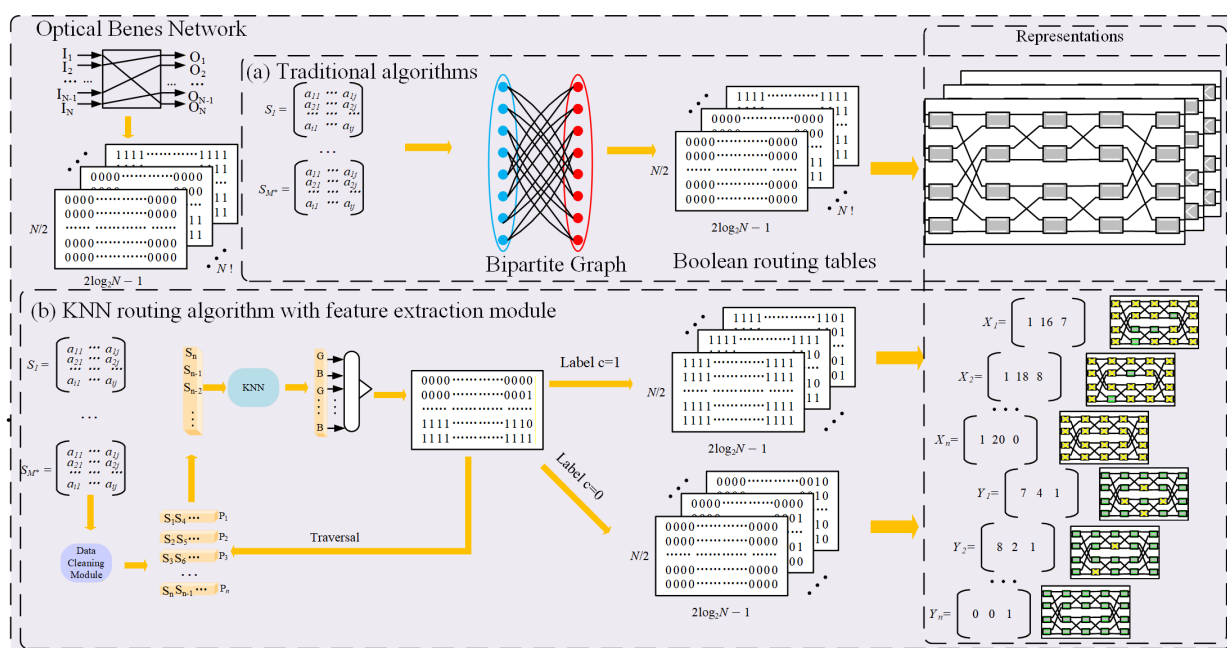


Figure 1. Comparison of routing algorithms and their representations. (a) Traditional routing algorithm based on bipartite graph matching. (b) Proposed KNN routing algorithm with a feature extraction module.

Figure 1b illustrates the core of the proposed KNN-based routing optimization algorithm with a feature extraction module, in which the characteristics of routing table information are effectively extracted via the KNN method to avoid selecting worst-case paths. Finally, a routing table with improved end-to-end EXT is selected by predicting the path quality from a representative subset, thereby efficiently bypassing the NP-hard $N!$ combinatorial solution space.

3.1. Conditions for Effective Use of KNN

The KNN algorithm is an instance-based, non-linear classification method relying on the Euclidean distance metric. Despite its simplicity, it can achieve high accuracy and structural stability when applied to properly selected and highly correlated feature spaces [25]. However, the practical application of the traditional KNN algorithm must meet the following three conditions simultaneously:

1. The Benes routing table contains $N!$ permutations, corresponding to $N!$ distinct data instances. For each classified data point, it is necessary to distinguish valid data from measurement noise. Notably, this noise is inherent to the data yet uncorrelated with the output configuration;
2. There is virtually no linear link between the columns in the routing table—a switch matrix of $N/2 \times (2 \log_2 N - 1)$ dimensions. On the other hand, each column (feature) needs to have a specific linear correlation;

3. The relationship between the hyperparameter K and the routing table distribution remains unclear. Nevertheless, determining the optimal value of K is critical to avoid overfitting or underfitting.

Therefore, to satisfy the basic requirements for applying KNN to Benes network routing tables and to address the aforementioned challenges, this paper proposes an improved KNN framework that incorporates a dedicated preprocessing and feature extraction stage.

3.2. Preprocessing Step

As mentioned earlier, the proposed KNN-based routing scheme must first satisfy the three requirements listed above to be applied to optical Benes networks. The specific preprocessing modules are as follows:

1. Data Cleaning Module: This module maps raw routing configurations to their corresponding valid $N!$ permutations.
2. Feature Extraction Module: This module converts the $N/2 \times (2\log_2 N - 1)$ switch matrix to a 1×3 feature vector. A significant advantage of this technique is its low computational cost in solving the linear correlation problem between columns.
3. K Selection Module: This module employs an exhaustive search to train each data group. The data group yielding the highest classification accuracy is used to determine the final value of K . Regarding the selection of K , exhaustive search is adopted because the application scenario of KNN is in the Benes optical chip network. Currently, the number of ports in this chip does not exceed 32. Therefore, the chosen K will be less than 32.

Through these preprocessing modules, the improved KNN algorithm is better able to handle the complexity of Benes network routing tables, ensuring high accuracy and efficiency in routing decisions.

3.3. Data, Training, and Testing Issues for the KNN Algorithm

The proposed routing algorithm must specify the following settings for data, training, and testing procedures to use KNN effectively:

1. Data Sources: Like other machine learning algorithms, KNN is a supervised learning algorithm that requires the generation of paired datasets for a specific problem. For each categorized data, metrics should be defined to distinguish between data and data noise. Similarly, criteria should be defined to obtain paired datasets.
2. Training Process: The criteria for selecting KNN inputs and the required accuracy should be defined.
3. Testing Process: In addition to accuracy, other transmission performance parameters should be defined.

These foundational data processing modules integrate seamlessly with the subsequent online inference stages to complete the framework of our proposed routing algorithm, as illustrated in Figure 2. The specific details of the algorithmic workflow are thoroughly discussed in Section 4.

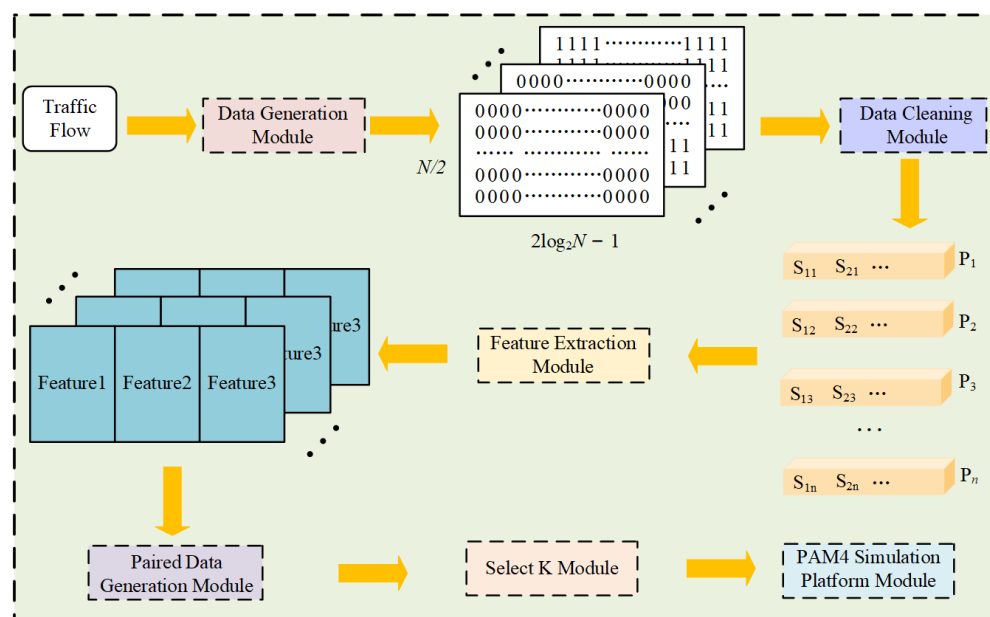


Figure 2. Overall framework of the proposed KNN routing algorithm, including data generation, cleaning, feature extraction, K-value selection, and PAM4 simulation modules.

4. Proposed KNN Routing Algorithm

For Benes networks, there is currently no extensive training dataset accessible in the literature. Thus, the problem of creating routing tables and allocating matching labels is handled by the Data Generation Module.

4.1. Data Generation Module

The status matrix, intrinsically comprising only Cross and Bar states, can be mapped to a Boolean matrix where ‘1’ and ‘0’ denote the respective states of the optical switches. As illustrated in Figure 3, this Boolean matrix is constructed via a dedicated data generation module. The module employs an iterative procedure to generate binary sequences of length $N/2 \times (2\log_2 N - 1)$, with each sequence representing a distinct routing configuration. Specifically, the process initializes a counter i , which iterates through the defined state space. During each iteration, the decimal index i is converted into a binary string of the required sequence length, corresponding to a unique switch configuration in the routing table. Through exhaustive iteration, this module compiles a comprehensive dataset encompassing all valid routing paths, thereby enabling the optical Benes network to dynamically reconfigure and route data in response to traffic demands.

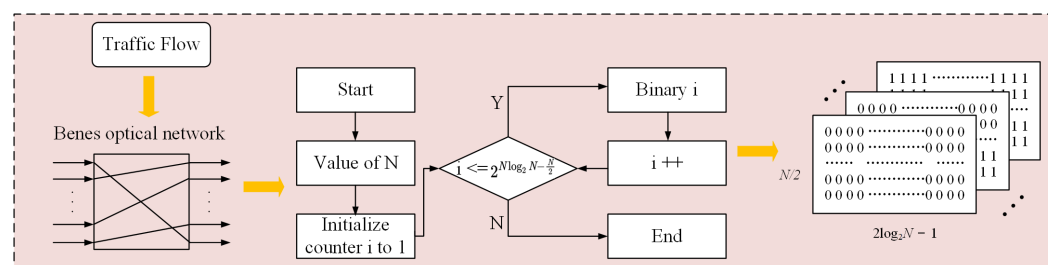


Figure 3. Flowchart of the Data Generation Module, which constructs the raw Boolean routing matrices by iteratively generating binary sequences.

4.2. Data Cleaning Module

Despite the massive volume of routing configurations, the Data Cleaning Module systematically assigns each input–output pair to one of the valid $N!$ permutation states.

In this work, each of the over 40,000 configurations is mutually uncorrelated with others; accordingly, data noise is strictly defined as redundant information irrelevant to valid permutation paradigms. The proposed module effectively eliminates such uncorrelated interference data, enabling rapid screening of valid routing pairs and retrieving complete routing matches within an extremely limited time.

The traditional routing algorithm uses the bipartite graph algorithm in the Benes network to randomly compute an output routing table for any input permutation as a solution to the matching problem. However, the reverse process of calculating the input routing table from the output is complex and time-consuming. As the number of ports increases, the calculation of each table requires polynomial time, which is costly.

To quickly obtain a large number of valid routing tables, we use a direct sequence exchange algorithm to compute the output matching for a randomly generated routing table, dividing it into $N!$ permutations. Owing to the known topology of the Benes network and the known switch states, this method can obtain a large number of valid routing tables in a shorter time.

Direct Sequence Exchange Algorithm: The main purpose of the direct sequence exchange algorithm is to randomly generate routing tables $S_n = a_{ij}$, while calculating the number of output matching permutations $P = P_1, P_2, \dots, P_{N!}$ and the count of each permutation corresponding to $P_j = S_{1j}, S_{2j}, \dots, S_{nj}$, as shown in Figure 4. Here, S_n represents the routing table digitized in binary form, while a_{ij} denotes the state of the optical switch in row i and column j . P encompasses all permutations of the routing table, where P_n represents the n th permutation. In the direct sequence exchange algorithm, the main focus is on how to categorize routing tables into $N!$ permutations, as shown in Algorithm 1.

Algorithm 1 Direct Sequence Exchange algorithm

Input: Operation sequence S_n and length of the list N

Output: A dictionary of lists and their corresponding operation sequences

custom_shuffle(list): The purpose of this function is to first split the list and then recombine the two resulting parts.

inverse_shuffle_sort(list): This function accepts a list *input_list* and returns a new list where elements are rearranged in an alternating pattern of odd and even indexed items.

custom_operation(list, values): This function swaps each pair of adjacent elements in the data list, based on the instructions in *values*. ('0' means no change; '1' means swap. The first element is swapped with the second, the third with the fourth, and so on.)

```

1: Initialize counter  $n \leftarrow 1$ 
2: while  $n \leq 2\log_2(N) - 1$  do
3:   Create a list  $L \leftarrow [1, 2, 3, \dots, N]$ .
4:   if  $n \leq \log_2(N - N/2)$  then
5:     Divide List  $L$  into  $\min(2^{n-1}, N/4)$  parts.
6:     Every List  $L_i = \text{custom\_shuffle}(L_i)$ .
7:   else
8:     Divide List  $L$  into  $\min(N^2/2^{n-1}, N/4)$  parts.
9:     Every List  $L_i = \text{inverse\_shuffle\_sort}(L_i)$ .
10:  end if
11:  Combine every list  $L_i$  into a NewList.
12:  Select Values =  $S_n$ .
13:  List  $L = \text{custom\_operation}(\text{NewList}, \text{Values})$ .
14:  Check the dictionary for a key with the same list  $L$ .
15:  if found then
16:    Add  $S_n$  to it.
17:  else
18:    Create a new key and add  $S_n$  to it.
19:  end if
20:   $n \leftarrow n + 1$ 
21: end while

```

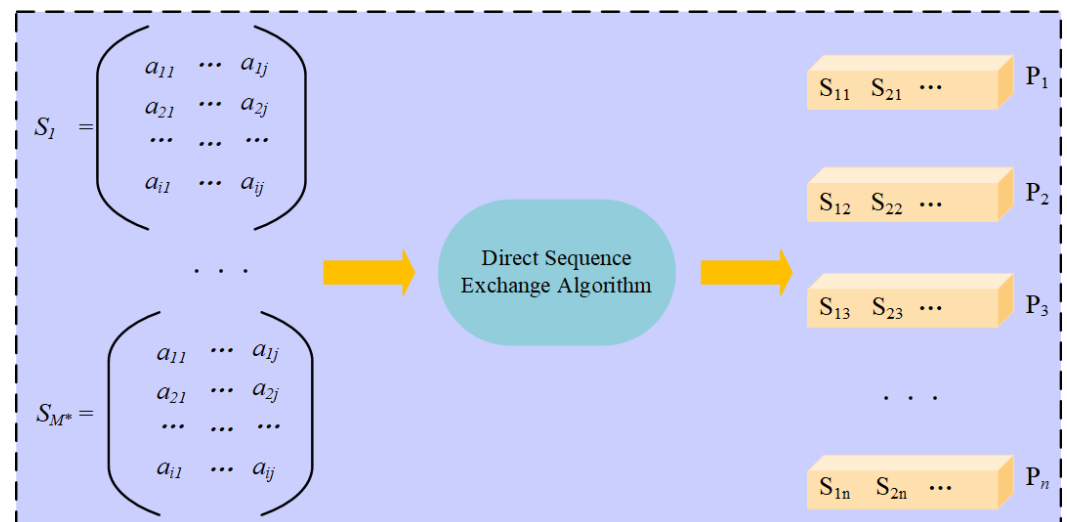


Figure 4. Workflow of the Data Cleaning Module, utilizing the Direct Sequence Exchange Algorithm to map raw switching configurations into valid permutations.

4.3. Feature Extraction Module

For the initialization, a counter n is defined and set to an initial value of 1. Subsequently, an integer list is defined, with elements incrementing from 1 to N , where N is a positive integer representing the number of ports in the optical Benes network. The counter n and the integer list represent the sequence number of the routing table and the destination list for the matching algorithm, respectively.

The module comprises a sequence of switch exchanges, conditional judgments, and path exchange operations. Specifically, the inverse perfect shuffle and forward perfect shuffle topological mappings govern the path exchange criteria, while a dedicated state evaluation logic determines the transition conditions for individual switches. Exploiting the inherent mirror symmetry of the Benes architecture, these dual mapping rules are essential for accurately resolving the routing configurations across both the front and rear stages of the network.

Finally, the module checks a dictionary to determine whether there is a key that matches the currently processed list. If there is, the algorithm adds S_n (the routing table of this iteration) to the value corresponding to that key; if not, a new key is created, and S_n is added. The counter n is then incremented by 1, and the process is repeated until the algorithm concludes. The vectorized feature extraction $X_n = f_1, f_2, f_3$ is the result of the feature extraction module, which takes as input the switching matrix of the Boolean routing table $S_n = a_{ij}$. As previously stated, our input data must simultaneously meet two of the three conditions mentioned above to be used with the KNN model: (1) The input data must be directly related to the routing table. (2) The input data matrix must have a linear correlation between columns, as indicated by the correlation coefficient.

Figure 5 illustrates two approaches to routing-table processing: (1) The direct segmentation method is shown in Figure 5a. It chooses the three highest correlation coefficients of the routing table column vector f_1, f_2, f_3 as a classification feature. (2) The feature extraction module is shown in Figure 5b. First, we convert the N -row by $(2 \log_2 N - 1)$ -column S_n into a matrix. Then, we extract three physical impairment features $X_n = f_1, f_2, f_3$. The extraction maps the discrete Boolean states into a continuous three-dimensional metric space.

In this metric space, the magnitude of the Euclidean distance itself does not directly reflect the absolute severity of physical impairments. Instead, it serves as a mathematical measure of similarity, demonstrating that the three extracted indicators (f_1, f_2, f_3) are sufficient to distinguish between superior and inferior routing configurations, while the

KNN algorithm validates the effectiveness of this feature extraction for classification, it cannot quantify which specific impairment plays the decisive role. To clarify the dominant mechanism, we additionally calculated the Pearson correlation coefficients between the extracted features and the final signal performance [26]. The results indicate that among the three extracted features, f_2 exhibits the most significant negative correlation with the Extinction Ratio (EXT) ($r = -0.4610$, $p < 0.001$), confirming it as a primary contributing physical factor.

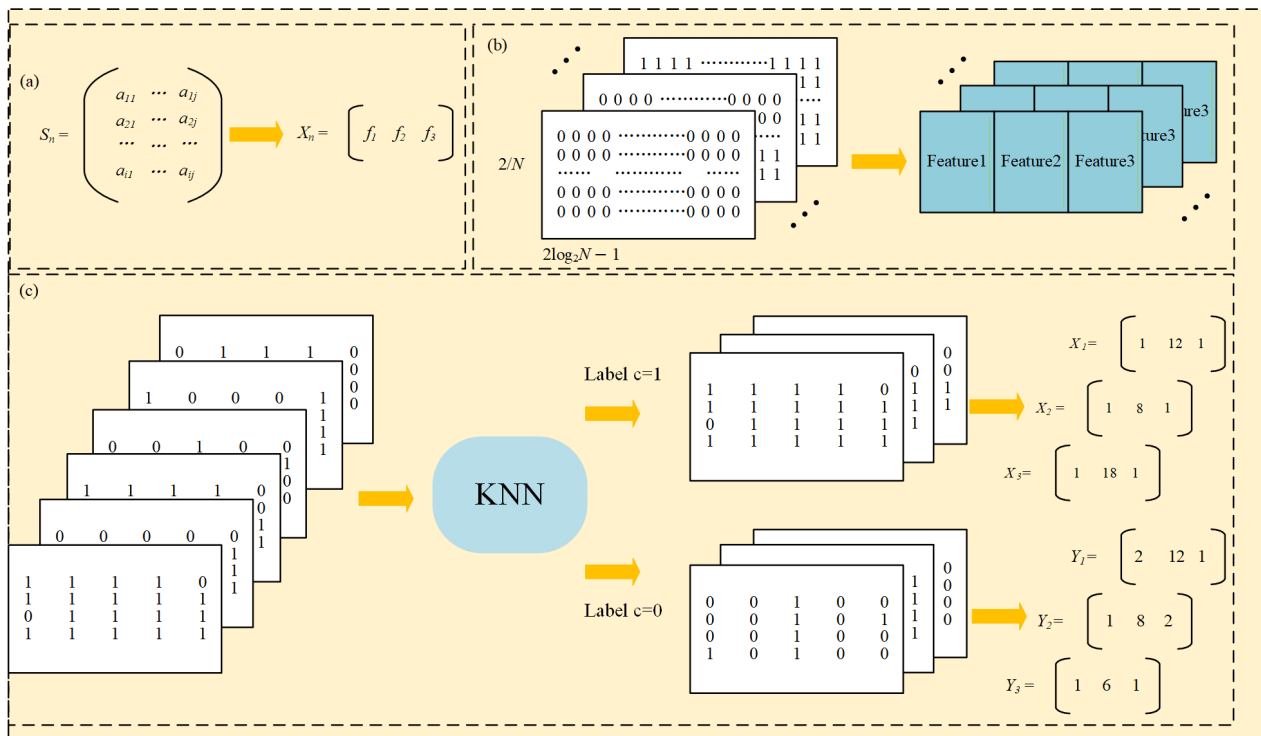


Figure 5. The Feature Extraction Module. (a) Extraction of local features (f_1, f_2, f_3) from the Boolean matrix. (b) Transformation into a 3D feature space. (c) KNN classification labeling for superior ($c = 1$) and inferior ($c = 0$) paths.

Inherently, the correlation between the columns of zeros and ones—representing the Bar and Cross states, respectively—in a raw Benes routing table is naturally very low, typically yielding an initial correlation coefficient of less than 0.12. This is fundamentally because the columns operate as independent routing switch stages; achieving a high correlation would mathematically require over 70% of the switching operations to execute identical “through” paths. Given the inherent randomness of dynamic traffic flows, such a scenario is practically impossible. Consequently, relying on raw Boolean data limits the classifier’s ability to capture the underlying physical deterioration, necessitating the proposed feature extraction module. By contrast, the proposed feature extraction bridges this gap. Because the extracted features (f_1, f_2, f_3) implicitly reflect the fundamental physical mechanisms of the Benes topology (i.e., crosstalk, cascaded loss, and path dependency), this topology-driven approach ensures that the resulting distance metric space consistently satisfies the requirements of the KNN model across different network sizes, data splits, and sampling strategies, provided that the underlying physical laws of the optical components remain constant.

Table 1 compares KNN models with and without the feature extraction module. Here, K varies from 1 to 7, and the test set proportion ranges from 10% to 40%. The findings indicate that prediction accuracy peaks when K is 5 and the test set:training set ratio is 3:7. As shown in Table 1, it is challenging to meet the 70% accuracy requirement without the

feature extraction module owing to the poor correlation coefficient. In contrast, adding the preprocessing module increases the accuracy by 8%, bringing it to 72.85%.

Table 1. Comparison of KNN models before and after the use of the feature extraction module.

Percentage of Tests (%)	Before Use of Feature Extraction Module				After Use of Feature Extraction Module			
	Accuracy (%)							
	$k = 1$	$k = 3$	$k = 5$	$k = 7$	$k = 1$	$k = 3$	$k = 5$	$k = 7$
10	64	64	64	64	54.28	62.85	67.25	67.14
20	37.25	41.17	43.13	52.94	68.57	68.57	70.71	70.71
30	52.63	63.15	64.47	56.57	63.50	71.50	72.85	65.40
40	55.88	51.96	50	47.05	61.56	64.41	61.92	53.73

4.4. Paired Data Generation Module

To train the model, a paired data generation module, as shown in Figure 6, must be incorporated. The optical Benes network has M potential routing tables, encompassing a complete Boolean state space from all zeros to all ones. For a network scale of $N = 8$, the state space M expands to approximately one million configurations. Because exhaustive testing of every configuration is computationally prohibitive, the routing tables are uniformly sampled at an interval of 100. Consequently, 10,000 samples are extracted to form the training and testing datasets.

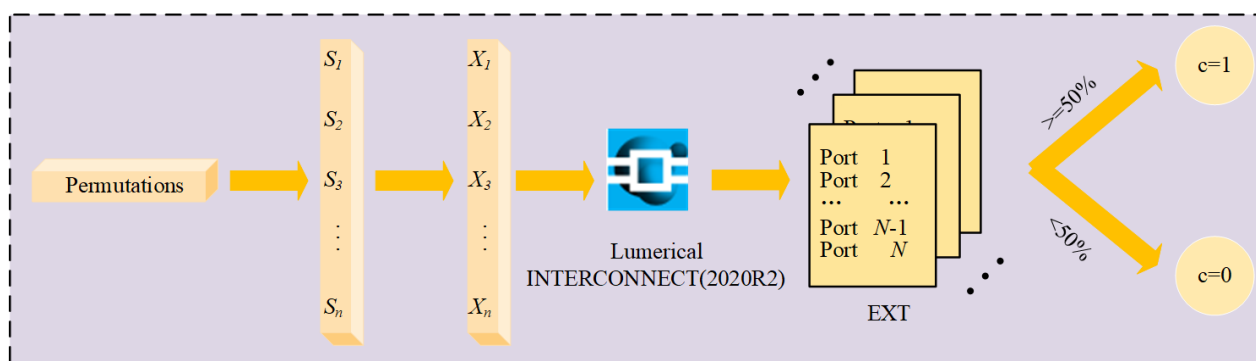


Figure 6. The Paired Data Generation Module, illustrating the EXT evaluation via Lumerical INTERCONNECT and the subsequent binary labeling process.

As illustrated in Figure 7, a comparison between the distributions of the full dataset and the sampled dataset demonstrates that the sampled subset largely preserves the statistical characteristics of the original data. This confirms the effectiveness and representativeness of employing a uniform sampling strategy when dealing with the massive routing state space of optical Benes networks.

The conventional algorithm typically partitions the routing tables into multiple domains. For a given input matching $P_j = \{S_{1j}, S_{2j}, \dots, S_{mj}\}$, it extracts local features $\{X_{1j}, X_{2j}, \dots, X_{mj}\}$ and assigns corresponding labels $c = \{c_{1j}, c_{2j}, \dots, c_{mj}\}$ for precise categorization. Here, the indices i and j are bounded by $1 \leq i < m$ and $1 \leq j \leq n$, respectively. The parameter n denotes the total number of domains, given by $n = N!$. The number of matchings is represented by m , while the total state space is defined as $M = 2^{(N/2) \times (2 \log_2 N - 1)}$.

The Lumerical INTERCONNECT(2020R2) software platform is then used to test the optical paths obtained from each routing table. The data are averaged and sorted from smallest to largest, with EXT as the primary metric. For example, $c_i = 1$ indicates that X_i is superior, whereas $c_i = 0$ indicates that X_i is inferior. The top 50% of the optical paths are

designated as "superior" with $c = 1$, while the bottom 50% are defined as "inferior" with $c = 0$.

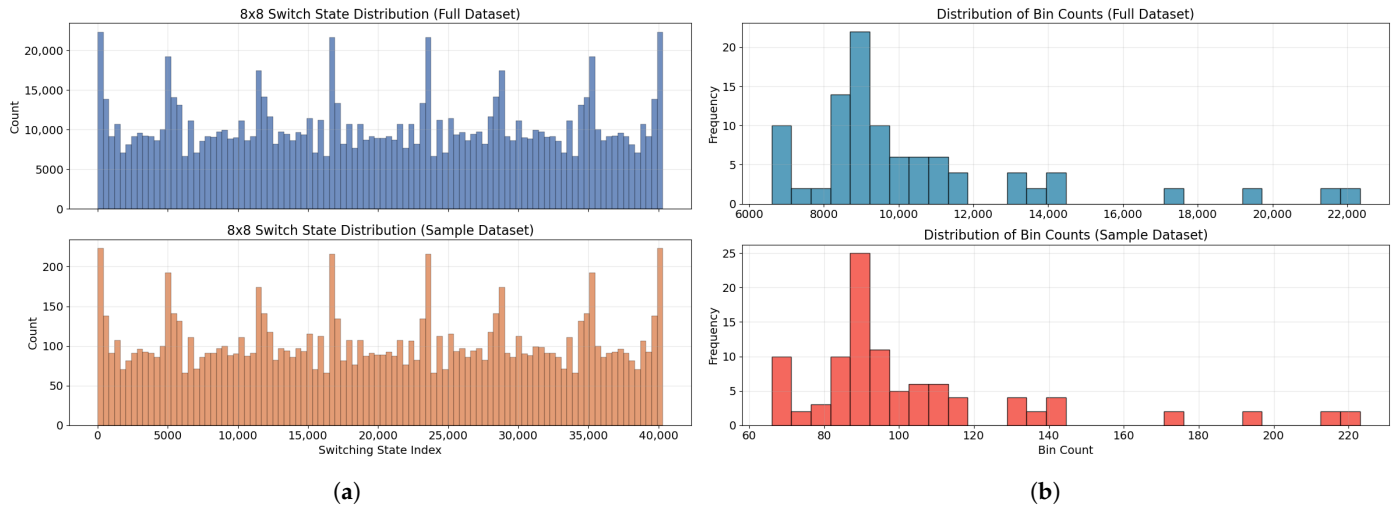


Figure 7. Statistical evaluation of the sampled routing dataset. (a) Occurrence distribution across switching state indices. (b) Frequency distribution of the switching state counts.

4.5. K Selection Module

The KNN algorithm considers the K nearest neighbors for any decision-making. After feature dimension reduction, we obtain three local features: $X_i = \{f_{1i}, f_{2i}, f_{3i}\}$, $X_i \in X$, and $X_j = \{f_{1j}, f_{2j}, f_{3j}\}$, $X_j \in X$. This module takes $\{X_j\}$ feature vectors as inputs and produces outputs with appropriate labels (c_j -labels) as shown in Figure 6.

Prior to KNN classification, all three extracted features are normalized to values in $[0, 1]$. This is defined as follows:

$$f'_{mi} = \frac{f_{mi} - f_{m,\min}}{f_{m,\max} - f_{m,\min}} \quad (1)$$

where f_{mi} represents the original value of the m th feature for sample i , $f_{m,\min}$ and $f_{m,\max}$ represent the global minimum and maximum values of the m th feature across the entire dataset, and f'_{mi} is the corresponding normalized feature value.

The Euclidean distance between two normalized feature vectors is calculated as

$$d(X_j, X_i) = \sqrt{\sum_{m=1}^3 (f'_{mj} - f'_{mi})^2} \quad (2)$$

where m represents the feature dimension ($m = 1, 2, 3$), f'_{mj} represents the normalized m th feature of the test sample X_j to be classified, and X_i represents the i th already-classified training sample.

In the KNN decision rule, the predicted category $\hat{c}_j$ of the test sample X_j depends on its distance d to the training samples. The K nearest samples, with known categories $c_{1j}, c_{2j}, \dots, c_{Kj}$, are used for voting, expressed as

$$\hat{c}_j = \text{mode}(c_{1j}, c_{2j}, \dots, c_{Kj}) \quad (3)$$

where c_{kj} represents the category of the k th nearest neighbor to the test sample X_j , while mode represents the most frequent category (i.e., the category with the highest number of occurrences).

To illustrate the relationship between the sample category and distance, consider X_j as a sample to be categorized. Its distances from the training samples are in ascending order $d_{1j}, d_{2j}, \dots, d_{Kj}$, with corresponding categories $c_{1j}, c_{2j}, \dots, c_{Kj}$. This can be expressed as

$$\hat{c} = \text{mode}(c_{kj} \mid d(X_j, X_i)), \text{ for } i = 1, 2, \dots, r \quad (4)$$

where r is the total number of training samples.

Once the predicted category $\hat{c}$ is determined, the model's overall performance must be quantitatively evaluated. Specifically, the classification accuracy of the proposed KNN routing algorithm is defined as the proportion of routing configurations correctly classified into their true categories, i.e., superior or inferior optical paths, out of all the configurations evaluated. Assuming there are n total routing tables to be classified, let X_j , where $j = 1, 2, \dots, n$, represent the j th routing table with its true label $c_j \in \{0, 1\}$. Here, 1 indicates a superior path and 0 indicates an inferior path. If n' represents the total number of tables where the predicted label exactly matches the true label, i.e., $\hat{c}_j = c_j$, the classification accuracy can be expressed as

Once the predicted category $\hat{c}_j$ is determined, the model's overall performance must be quantitatively evaluated. Specifically, the classification accuracy of the proposed KNN routing algorithm is defined as the proportion of routing configurations correctly classified into their true categories, i.e., superior or inferior optical paths, out of all the configurations evaluated. Assuming there are n total routing tables to be classified, let X_j , where $j = 1, 2, \dots, n$. It represents the j th routing table with its true label $c_j \in \{0, 1\}$. Here, 1 indicates a superior path and 0 indicates an inferior path. Let $\hat{c}_j$ denote the predicted category for X_j using the KNN decision rule. If n' represents the total number of tables where the predicted label exactly matches the true label, i.e., $\hat{c}_j = c_j$, the classification accuracy can be expressed as

$$\text{Accuracy} = \frac{n'}{n} = \frac{1}{n} \sum_{j=1}^n I(\hat{c}_j = c_j) \quad (5)$$

where $I(\cdot)$ is the indicator function, which equals 1 if the condition $\hat{c}_j = c_j$ is true and 0 otherwise.

An appropriate K value is essential to the algorithm's performance. Although it may result in overfitting, a lower K value increases sensitivity to local features. Conversely, higher K values limit error and reduce noise, which may result in an oversimplified model that ignores the data's intrinsic complexity [27]. The selection of K values in this paper is restricted to odd numbers between 1 and 19. Only odd numbers are chosen for K to avoid voting ties in binary classification.

Figure 8 shows how various test set proportions, from 10% to 40%, affect the accuracy of the proposed KNN algorithm. The findings show that the proposed KNN algorithm achieves a maximum accuracy of 72.85% when $K = 5$, surpassing 70% accuracy for every test set proportion. The accuracy remains above 65% for all four test set proportions (q denotes the fraction of the test set) except $K = 1$, where the accuracy decreases. In comparison, the accuracy is only about 55% without feature extraction. We then confirm the transmission performance of the proposed KNN routing algorithm using optical simulation software.

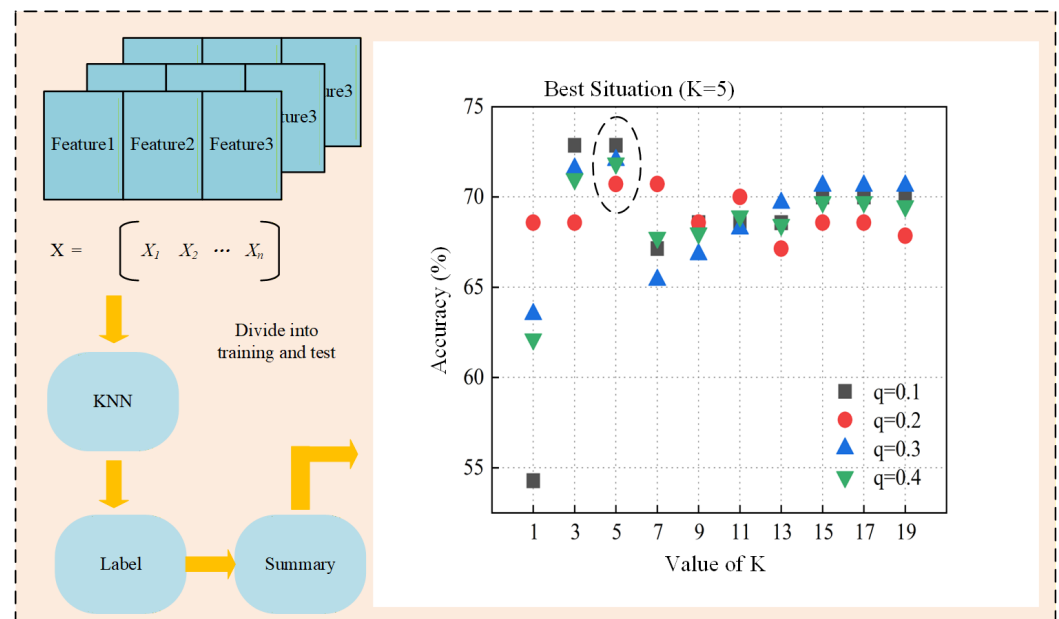


Figure 8. Impact of the hyperparameter K on the classification accuracy of the KNN model under different test set proportions ($q = 0.1$ to $q = 0.4$).

5. Simulation Results

5.1. PAM4 Simulation Test Module

The PAM4 simulation platform used in this paper is shown in Figure 9. Two PRBS-15 generators produce a 30 Gbps electrical signal. An MZI modulator loads the two sets of orthogonally phased electrical signals onto a wavelength of 1550 nm. The Benes optical switch is then used to switch the optical signals to the appropriate output ports. A photodetector (PD) at the receiving end transforms the optical signal into an electrical signal after it has passed through an attenuator. Physical impairments such as the Symbol Error Rate (SER) are observed by comparing the input and output quadrature bit streams. Table 2 lists the comprehensive simulation parameters.

Table 3 compares the performance of our proposed KNN routing algorithm with state-of-the-art CNN and SVM techniques. To ensure a fair assessment, all models were evaluated under strictly identical conditions: they shared the same dataset of 10,000 configurations with identical training-to-test split ratios, and standard hyperparameter optimization procedures were applied to the baseline models.

To align with standard practices, both the CNN [28] and SVM [29] models utilize the original Boolean routing table matrices as their baseline inputs. In contrast, our KNN model processes the same 10,000 routing configurations through our proposed feature extraction module, utilizing the resulting three extracted features as its input, which constitutes the core innovation of this work. Therefore, rather than merely comparing isolated classifiers, this setup rigorously evaluates our proposed lightweight framework against conventional deep learning and classical pipelines under an identical raw data baseline.

Evaluated on the same test dataset, the proposed KNN routing algorithm achieves a maximum accuracy of 72.85%, significantly outperforming the SVM (with an accuracy of 51.06%) and the CNN (with 52.23%). Given the immense state space of the routing problem, exceeding the 70% accuracy threshold demonstrates that the proposed KNN method is highly competent for this classification task.

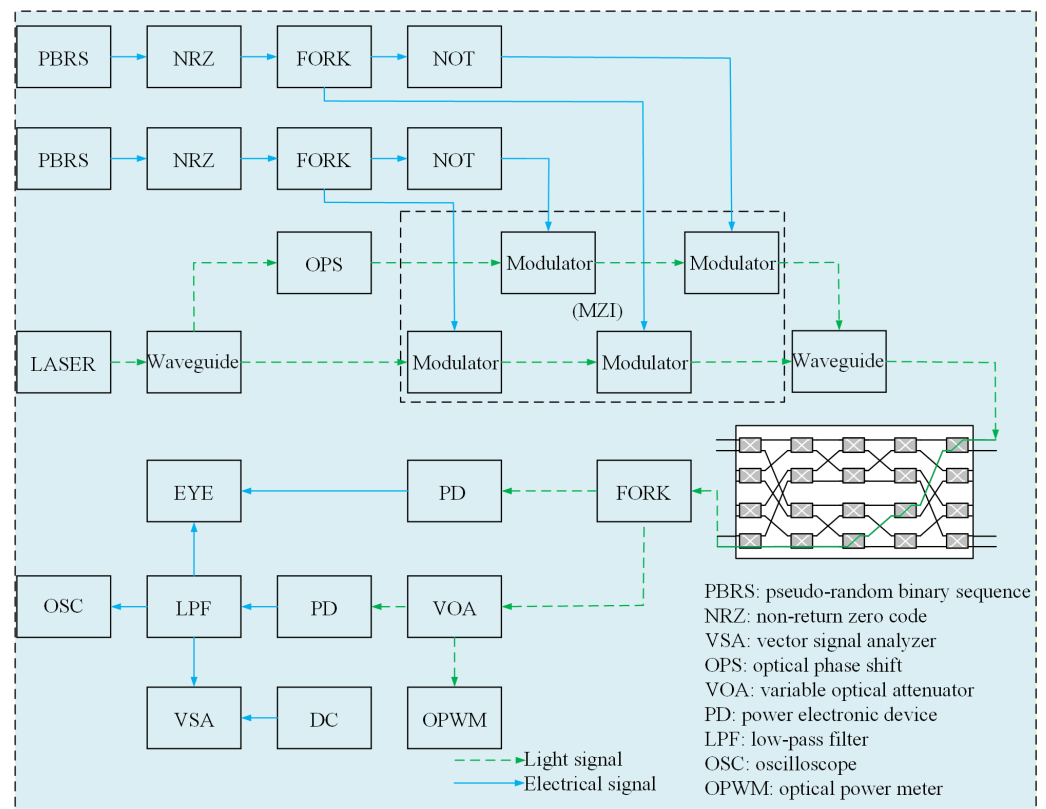


Figure 9. Schematic of the PAM4 simulation test platform used to evaluate the transmission performance of the optical Benes network.

Table 2. Simulation parameters.

Parameter	Value
Size of network	8, 16, 32 ports
Modulation scheme	PAM4
Data rate	30 G
Operating wavelength	1550 nm
Extinction Ratio (EXT)	>28 dB
SER	10^{-3} to 10^{-9}
Iterations	10,000
Received optical power	− 15 dBm
Switching time	0.0041 μ s

Table 3. Comparison of KNN against the state-of-the-art CNN and SVM methods.

Method	Test Accuracy	Computation Cost (per Iteration)
CNN	52.23%	5–10 ms
SVM	51.06%	1–2 ms
KNN (Proposed)	72.85%	0.1–1 ms

The findings demonstrate that the proposed KNN routing method achieves the best accuracy with the lowest computational cost. Algorithmically, this superior performance stems from the nature of the extracted feature space. The three extracted features (f_1, f_2, f_3) map physical impairments into localized, dense clusters that are highly compatible with KNN’s distance-based metric. In contrast, SVM attempts to separate classes via hyperplanes, which proves ineffective against the complex, highly non-linear boundaries of superior and inferior paths. Furthermore, CNN operates as a local filter, hierarchically extracting spatial features from low to high levels before achieving classification through

fully connected layers. However, owing to the highly repetitive topological structure of the Benes architecture, CNN struggles to learn distinguishable features directly from these image-like representations, resulting in degraded accuracy. Finally, as Table 3 indicates, the proposed approach directly invokes the KNN algorithm for categorization, yielding the shortest execution time.

Because our approach directly invokes the KNN algorithm for path categorization, the computational complexity of our routing strategy equals that of the KNN model, which is $O(M \times D)$. Here, M is the number of training samples (a predefined proportion of our 10,000 total samples), and D is the number of features ($D = 3$). With these fixed constants, our algorithm significantly bypasses the high computational complexity commonly associated with conventional Benes routing algorithms.

5.2. Simulation EXT Results

Figure 10 summarizes the port EXT comparison before and after the use of the optimized routing table with specific switch configuration settings. Owing to the random path selection of the traditional routing table, the output optical power was lower and fluctuated significantly before the implementation of the proposed KNN routing technique. This is illustrated in Figure 10, where the lower EXT value marked by the red circle represents the conventional routing method and its higher fluctuation as the switch configuration changes.

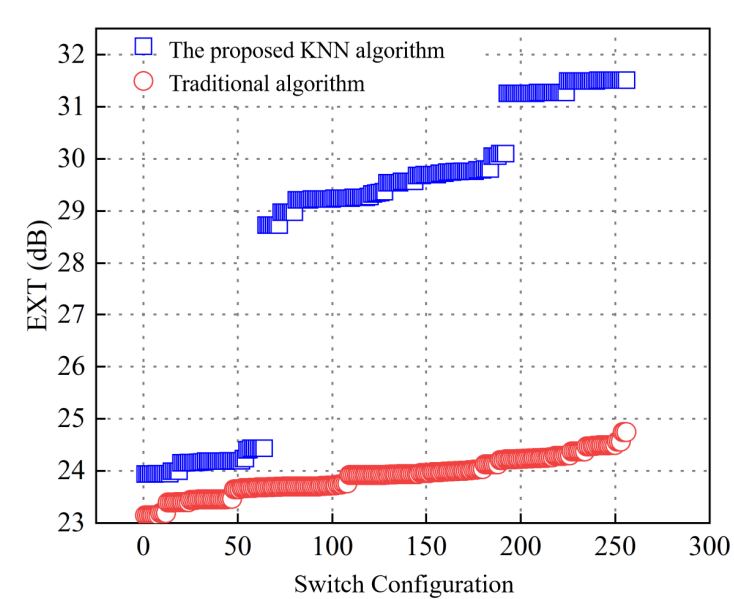


Figure 10. Comparison of port EXTs of the proposed KNN algorithm and the traditional algorithm.

Nevertheless, the output optical power is greatly improved and the volatility is significantly reduced after path optimization using the proposed KNN routing technique. This improvement is also illustrated in Figure 10. In general, the EXT values are higher with the proposed algorithm than with the conventional routing algorithm. The upward trend becomes smoother as the number of switch configurations increases.

5.3. Simulation SER Results

Figure 11 illustrates the relationship between the $\log(\text{SER})$ and the received optical power. At the target SER threshold, the minimum received optical power required by the proposed KNN routing technique is significantly improved to -8.78 dBm. This represents an improvement of approximately 0.83 dB compared to the -7.95 dBm required by the conventional method. It should be noted that these curves are generated from the average

of five independent experimental trials; explicit statistical uncertainty is not provided here owing to the limited sample size.

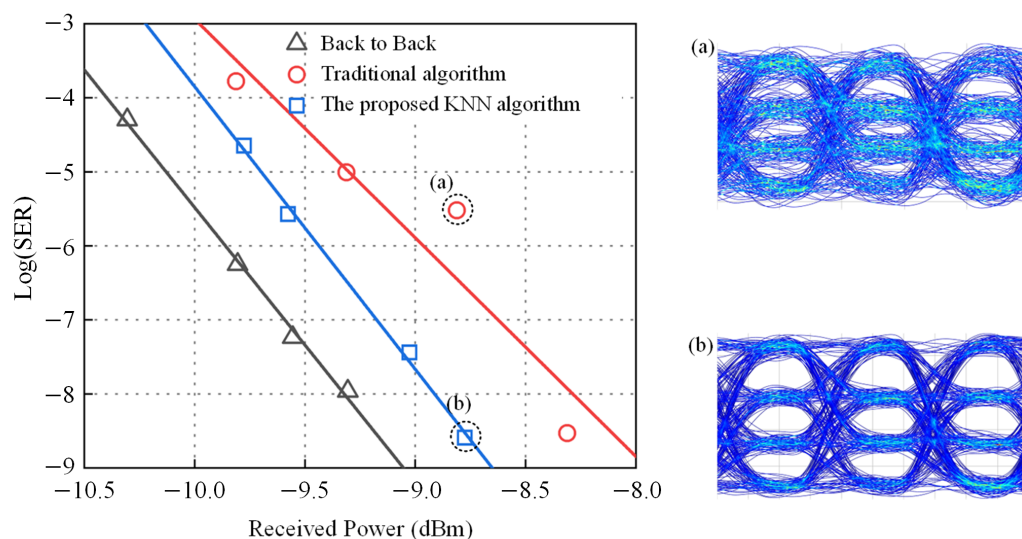


Figure 11. Comparison of port Symbol Error Rates (SERs) vs. received optical power. Insets: simulated eye diagrams at -8.6 dBm for (a) the traditional algorithm and (b) the proposed KNN algorithm.

All-channel data show that the proposed KNN routing technique successfully preserves excellent signal quality. As seen in Figure 11a the eye diagram of the conventional method is nearly closed with the received power at -8.6 dBm. However, as Figure 11b illustrates, the eye diagram of the proposed KNN algorithm is suitably open at the same received power.

6. Discussion

OSNR, BER, and EXT are three crucial technical indicators essential for assessing the performance of transmission paths [16–19]. Because determining the EXT takes a comparatively small amount of time, this study employs it rather than OSNR and BER. As shown in Table 4, the EXT calculation takes only 4 s—an order of magnitude less time than OSNR and BER. Each trial involves the computation of massive routing components. Without GPU acceleration, a single trial requires at least several days to complete. Furthermore, our experiment encompasses one million independent trials, which would require approximately half a year for full data collection, resulting in extremely low time efficiency.

Table 4. Time penalty comparison.

Performance Indicator	Time Penalty (per Iteration)	Total Time (Day)
EXT	4 s	48.4
BER	17 s	205.6
OSNR	22 s	267.6

Consequently, we consider EXT to be the most suitable central parameter. The primary distinctions between the proposed KNN algorithm and alternative routing algorithms are shown in Table 5. In particular, PILOSS networks are covered in studies [30,31], while mesh networks are covered in studies [32,33]. Most of these studies use routing methods, and references [34,35] as well as [36–41] focus on Benes networks.

Table 5. Comparison with state-of-the-art methods.

References	Topology	Manual Selection	Routing Algorithm	Machine/Deep Learning	Feature Extraction	Feature Interpretation	Year
[30]	PILOSS	–	–	–	✓	–	2016
[31]	PILOSS	✓	–	–	–	–	2014
[32]	Mesh	✓	✓	–	–	–	2012
[33]	Mesh	–	✓	✓	✓	–	2024
[34]	Benes	✓	✓	–	✓	–	2022
[35]	Benes	✓	✓	–	–	–	2024
[36]	Benes	–	✓	–	–	–	2019
[37]	Benes	✓	–	–	–	–	2021
[38]	Benes	✓	–	–	–	–	2023
[39]	Benes	–	✓	✓	✓	–	2023
[40]	Benes	–	–	✓	–	–	2020
[41]	Benes	✓	–	–	–	–	2022
Our Work	Benes	–	–	✓	✓	✓	2026

When applied to Benes networks, our approach offers two crucial advantages. First, the proposed KNN technique outperforms previous ML/DL algorithms, since the feature extraction module lowers the likelihood of worst-case connections. Second, the proposed KNN routing technique enhances feature interpretability, making the feature extraction procedure easier for humans to understand. In the existing literature, references [30,34] examine feature extraction, whereas [33,39] investigate ML and DL methods. Most of the time, though, the extracted features are not interpreted.

7. Conclusions

This paper proposed a KNN routing algorithm with a feature extraction module for optical Benes networks to enhance accuracy in predicting worst-case paths. Traditional methods are often exhaustive and involve substantial computational overhead. Training the KNN model addresses the routing table screening problem as a distance-based classification problem, allowing the prediction of subsequent path states from a subset of the data. This approach has the potential to reduce screening time and improve EXT. The improvement in transmission performance is validated using a PAM4 simulation platform, focusing on horizontal and vertical metrics, BER, and power consumption. The model achieves an accuracy of 72.85%, establishing it as a practical algorithm. Simulation results demonstrate that 30 Gbps end-to-end communication is achieved. Additionally, the minimum required optical power for error-free transmission exhibits a 0.83 dB advantage for the optimal path over the worst-case path, confirming the reduced optical power penalty and superior link budget achieved by the proposed KNN routing algorithm.

Author Contributions: Conceptualization, L.Z. and B.H.; methodology, L.Z.; software, B.H.; validation, L.Z. and B.H.; formal analysis, L.Z.; investigation, L.Z. and Y.K.; resources, L.Z.; data curation, B.H.; writing—original draft preparation, L.Z. and B.H.; writing—review and editing, B.H., S.B.H. and A.S.; visualization, B.H.; supervision, L.Z. and S.B.H.; project administration, L.Z.; funding acquisition, L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (62201338).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data generated and analyzed in this study are available from the corresponding author upon reasonable request.

Acknowledgments: This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62201338.

Conflicts of Interest: Li Zhao was employed by the Anhui Zhiguo Intelligent Technology Co., Ltd. The authors declare no conflicts of interest.

References

- Pagani, S.; Manoj, P.S.; Jantsch, A.; Henkel, J. Machine learning for power, energy, and thermal management on multicore processors: A survey. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2018**, *39*, 101–116.
- Qiao, L.; Tang, W.; Chu, T. 32×32 silicon electro-optic switch with built-in monitors and balanced-status units. *Sci. Rep.* **2017**, *7*, 42306.
- Cheng, Q.; Bahadori, M.; Huang, Y.; Rumley, S.; Bergman, K. Smart routing tables for integrated photonic switch fabrics. In *2017 European Conference on Optical Communication (ECOC)*; IEEE: New York, NY, USA, 2017; pp. 1–3.
- Li, J.; Yun, Y.; Xu, K.; Zhang, J.; Lin, H.; Zhang, Y.; Hu, J.; Gu, T. Performance limits of phase change integrated photonics. *IEEE J. Sel. Top. Quantum Electron.* **2024**, *30*, 6100109.
- Guo, Z.; Lu, L.; Zhou, L.; Shen, L.; Chen, J. 16×16 silicon optical switch based on dual-ring-assisted Mach–Zehnder interferometers. *J. Light. Technol.* **2017**, *36*, 225–232.
- Li, X.; Gao, W.; Lu, L.; Chen, J.; Zhou, L. 8×8 microring optical switch on a dual-layer Si₃N₄-on-SOI platform. In *Asia Communications and Photonics Conference*; Optica Publishing Group: Washington, DC, USA, 2021; p. T1I-1.
- Lu, L.; Zhao, S.; Zhou, L.; Li, D.; Li, Z.; Wang, M.; Li, X.; Chen, J. 16×16 non-blocking silicon optical switch based on electro-optic Mach–Zehnder interferometers. *Opt. Express* **2016**, *24*, 9295–9307.
- Lu, L.; Li, X.; Gao, W.; Li, X.; Zhou, L.; Chen, J. Silicon non-blocking 4×4 optical switch chip integrated with both thermal and electro-optic tuners. *IEEE Photonics J.* **2019**, *11*, 6603209.
- Zulfin, M.; Fauzi, R.; Ananda, Y.; Pinem, M.; Razali, M. Performance Comparison of Looping and Input-Output Algorithms on the 8×8 Benes Switching Network. In *2019 3rd International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM)*; IEEE: New York, NY, USA, 2019; pp. 53–56.
- Fineman, J.T. Nearly work-efficient parallel algorithm for digraph reachability. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, Los Angeles, CA, USA, 25–29 June 2018; pp. 457–470.
- Konoike, R.; Suzuki, K.; Kawashima, H.; Ikeda, K. Port-alternated switch-and-select optical switches. *J. Light. Technol.* **2020**, *39*, 1102–1107.
- Tanizawa, K.; Suzuki, K.; Ikeda, K.; Namiki, S.; Kawashima, H. Non-duplicate polarization-diversity 8×8 Si-wire PILOSS switch integrated with polarization splitter-rotators. *Opt. Express* **2017**, *25*, 10885–10892.
- Nikolova, D.; Rumley, S.; Calhoun, D.; Li, Q.; Hendry, R.; Samadi, P.; Bergman, K. Scaling silicon photonic switch fabrics for data center interconnection networks. *Opt. Express* **2015**, *23*, 1159–1175.
- Zhang, S.; Cheng, D.; Deng, Z.; Zong, M.; Deng, X. A novel kNN algorithm with data-driven k parameter computation. *Pattern Recognit. Lett.* **2018**, *109*, 44–54.
- Yu, H.; Hutson, A.D. A robust Spearman correlation coefficient permutation test. *Commun.-Stat.-Theory Methods* **2024**, *53*, 2141–2153.
- Lin, X.; Dobre, O.A.; Ngatched, T.M.; Eldemerdash, Y.A.; Li, C. Joint modulation classification and OSNR estimation enabled by support vector machine. *IEEE Photonics Technol. Lett.* **2018**, *30*, 2127–2130.
- Zhang, Q.; Yang, Y.; Guo, C.; Zhou, X.; Yao, Y.; Lau, A.P.T.; Lu, C. Accurate BER estimation scheme based on K-means clustering assisted Gaussian approach for arbitrary modulation format. *J. Light. Technol.* **2019**, *38*, 2152–2157.
- Wang, D.; Zhang, M.; Li, Z.; Li, J.; Fu, M.; Cui, Y.; Chen, X. Modulation format recognition and OSNR estimation using CNN-based deep learning. *IEEE Photonics Technol. Lett.* **2017**, *29*, 1667–1670.
- Zhang, Q.; Zhou, H.; Jiang, Y.; Cao, B.; Li, Y.; Song, Y.; Chen, J.; Zhang, J.; Wang, M. A simple joint modulation format identification and OSNR monitoring scheme for IMDD OOFDM transceivers using K-nearest neighbor algorithm. *Appl. Sci.* **2019**, *9*, 3892.
- Duran, B.; Emory, D.; Azam, Y.E.; Linzell, D.G. A novel CNN architecture for robust structural damage identification via strain measurements and its validation via full-scale experiments. *Measurement* **2025**, *239*, 115393.
- Gragopoulos, I.; Pavlidou, F.N. A new evaluation criterion for Clos-and Benes-type rearrangeable switching networks. *IEEE Trans. Commun.* **2002**, *45*, 119–126.
- Zhang, Q. On relationships between Chatterjee’s and Spearman’s correlation coefficients. *Commun. Stat. Theory Methods* **2025**, *54*, 259–279.
- Sun, Z.; Zhao, L.; Hussain, S.B.; Sultan, A.; Shi, X. Preprocessing convolutional neural networks based routing algorithm for PILOSS optical switching networks. *Opt. Express* **2024**, *32*, 42951–42963.
- Witkowski, C. A Turing Test for Self-Awareness. *Neurips 2024* **2024**, 13578.

25. Li, S.; Harner, E.J.; Adjero, D.A. Random KNN feature selection—a fast and stable alternative to Random Forests. *BMC Bioinform.* **2011**, *12*, 450.
26. Qin, X.; Li, J.; Sun, B.; Gu, Z.; Jiang, X.; Gao, X.; Gu, R.; Huo, X.; Ji, Y. Feature Correlation-based Data Augmentation for QoT Estimation in Optical Networks. In *2025 Asia Communications and Photonics Conference (ACP)*; IEEE: New York, NY, USA, 2025; pp. 1–4.
27. Li, W.; Wang, L.; Xu, J.; Huo, J.; Gao, Y.; Luo, J. Revisiting local descriptor based image-to-class measure for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 15–20 June 2019; pp. 7260–7268.
28. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90.
29. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*; IEEE: New York, NY, USA, 2005; Volume 1, pp. 886–893.
30. Tanizawa, K.; Suzuki, K.; Ikeda, K.; Namiki, S.; Kawashima, H. Novel polarization diversity without switch duplication of a Si-wire PILOSS optical switch. *Opt. Express* **2016**, *24*, 6861–6868.
31. Suzuki, K.; Tanizawa, K.; Matsukawa, T.; Cong, G.; Kim, S.H.; Suda, S.; Ohno, M.; Chiba, T.; Tadokoro, H.; Yanagihara, M.; et al. Ultra-compact 8 × 8 strictly-non-blocking Si-wire PILOSS switch. *Opt. Express* **2014**, *22*, 3887–3894.
32. Chawade, S.D.; Gaikwad, M.A.; Patrikar, R.M. Review of XY routing algorithm for network-on-chip architecture. *Int. J. Comput. Appl.* **2012**, *43*, 975–8887.
33. Shahane, P.; Kurup, R. Design of fault tolerant algorithm for network on chip router using field programmable gate array. *Int. J. Reconfigurable Embed. Syst.* **2024**, *13*, 1.
34. Zhao, L. Power-aware path selection routing algorithm for benes integrated photonic switches. In *Eighth Symposium on Novel Photoelectronic Detection Technology and Applications*; SPIE: Bellingham, Washington, USA, 2022; Volume 12169, pp. 1102–1107.
35. Zhao, L.; Li, Z.; Ma, T. Making path selection bright: A routing algorithm for on-chip benes networks. *Electronics* **2024**, *13*, 981.
36. Cheng, Q.; Huang, Y.; Bahadori, M.; Zhou, J.; Glick, M.; Bergman, K. Fabric-wide, penalty-optimized path routing algorithms for integrated optical switches. In *Optical Fiber Communication Conference*; Optica Publishing Group: Washington, DC, USA, 2019; p. Th3A-4.
37. Khope, A.S.; Samanta, A.; Xiao, X.; Yoo, B.; Bowers, J.E. Review of integrated photonic elastic wdm switches for data centers. *arXiv* **2021**, arXiv:2105.14934.
38. Cheng, Q.; Yao, C.; Calabretta, N.; Stabile, R.; Suzuki, K.; Kawashima, H.; Tang, W.; Glick, M.; Chu, T.; Ikeda, K.; et al. Photonic switch fabrics in data center/high-performance computing networks. In *Integrated Photonics for Data Communication Applications*; Elsevier: Amsterdam, The Netherlands, 2023; pp. 265–301.
39. Gostimirovic, D.; Soref, R. An integrated optical circuit architecture for inverse-designed silicon photonic components. *Sensors* **2023**, *23*, 626.
40. Kouissi, M.; Charbonnier, B.; Algani, C. Comprehensive model for evaluating the performance of Mach-Zehnder-based silicon photonic switch fabrics in large scale. *Appl. Sci.* **2020**, *10*, 8688.
41. Yin, Y.; Sun, C.; Ding, Y.; Xu, X.; Yao, M.; Li, L.; Lin, H.; Zhang, D. Scalable 3D silicon optical switch based on CLOS architecture. *IEEE Photonics J.* **2022**, *14*, 6645109.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.