IBIMA Publishing
Communications of International Proceedings
https://ibimapublishing.com/p-articles/44AI/2024/4451024/
Vol. 2024 (22), Article ID 4451024

Control of Switching Fabrics Using Artificial Intelligence*

[1]Piotr JASTRZEBSKI and [2]Mariusz ZAL

[1]Systell, Poznań, Poland

[2]Poznan University of Technology, Chair of Communication and Computer Networks, Poznań, Poland

Correspondence should be addressed to: Piotr JASTRZEBSKI, pawel.krzyzaniak@student.put.poznan.pl

* Presented at the 44th IBIMA International Conference, 27-28 November 2024 Granada, Spain

Abstract

Control of switching fabrics is a complex issue. Depending on the chosen algorithm, a switching fabric exhibits different combinatorial properties that impact its ability to provide services with a given Quality of Service (QoS). The complexity of the algorithm varies depending on the type of switching fabric and the desired properties (e.g., non-blocking in the narrow or wide sense, repackable or rearrangeable fabrics, or blocking fabrics). Algorithms also differ in how they affect the state of the switching fabric (and already established connections) when handling a new request.This article attempts to leverage artificial intelligence to control a switching fabric. Standard machine learning mechanisms were utilized for this purpose. The data used to train the models was derived from simulations of switching fabric operation controlled by known path selection algorithms. While for non-blocking fabrics, the AI model demonstrated connection losses (as expected, since these fabrics are designed to be lossless for specific path selection algorithms), for blocking fabrics, the AI model significantly outperformed traditional approaches in accepting new requests.

Keywords: machine learning, control algorithms, switching elements, switching fabric

Introduction

Currently, enterprises, institutions, and ordinary citizens, as users of electronic devices, send and receive vast amounts of information. For this process to run smoothly, they rely—consciously or not—on telecommunication networks and network devices. Switching networks enable the transmission of information in the desired direction using various algorithms, and they are integral components of these devices. This makes them a critical element in everyday life, allowing us to receive messages addressed specifically to us and to send them to a chosen recipient with near certainty that the message will be delivered.

There is a noticeable increase in the application of artificial intelligence in our lives, spanning various fields from marketing to medical diagnostics. One definition of artificial intelligence states that it is "a field of science attempting to explain and emulate intelligent behavior using computational methods" [1]. However, this definition does not provide much clarity. Broadly speaking, AI can be considered a branch of computer science focused on developing algorithms capable of adapting to changing conditions, making decisions, or engaging in abstract thinking [2].

A key component of artificial intelligence is machine learning, which involves processing large amounts of data and using it to develop algorithms, behavioral models, or decision-making systems. A distinctive feature of machine learning is the improvement of application performance through data processing. The application of such an approach to managing the operation of a switching network will be the subject of this work.

Cite this Article as: Piotr JASTRZEBSKI and Mariusz ZAL, Vol. 2024 (22) "Control of Switching Fabrics Using Artificial Intelligence" Communications of International Proceedings, Vol. 2024 (22), Article ID 4451024, https://doi.org/10.5171/2024.4451024

Using machine learning algorithms, the aim of this work is to create a path selection model for specified switching networks, address the theoretical aspects related to the topic, and compare the model's performance with switching network control algorithms.

The scope of this work includes gathering fundamental knowledge about switching networks and machine learning algorithms, preparing training data that will serve as the basis for creating a path selection model using a support vector machine. The creation of the model is the central element of this work, allowing for a comparison of its results with those obtained from path selection algorithms commonly used in practice.

Switching fabric

Switching fabric classyfication

The basic element of every switching network is a switch, which can have m inputs and n outputs. A switch where the number of inputs equals the number of outputs, i.e., of size m × m, is called a symmetric switch. Switches can be represented in various ways (Fig. 1). By connecting them in an appropriate way, we can construct a switching network (Fig. 2).

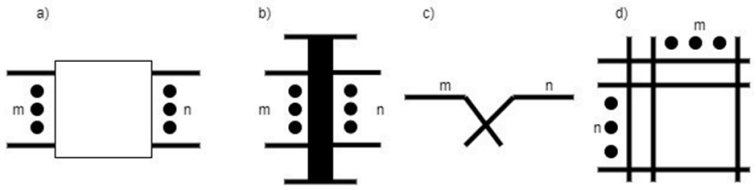

Figure 1. Symbols of commutators with dimensions m x n.

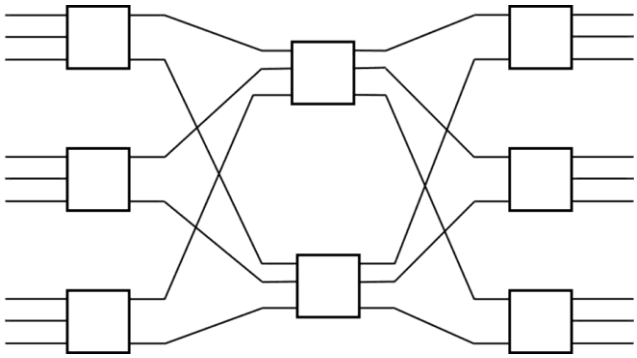

Figure 2. Switching field constructed from the connection of commutators.

Switching networks can be categorized in various ways, depending on the features and properties considered. The most general classification divides networks based on the method of call path distribution. This includes:

- Spatial networks,
- Temporal networks,
- Frequency-based networks.

In spatial networks, each connection is established on a physically separate path. In temporal networks, channels are switched by time shifts, whereas in frequency-based networks, each channel is assigned a different carrier frequency.

The earlier-mentioned interconnection of switches requires a classification of switching networks based on the number of stages, resulting in two options:

- Single-stage networks,
- Multi-stage networks.

If the desired connection is established through a single switching point, it is a single-stage network. If it involves connecting the output of one switch to the input of another, it is a multi-stage network (Fig. 3).

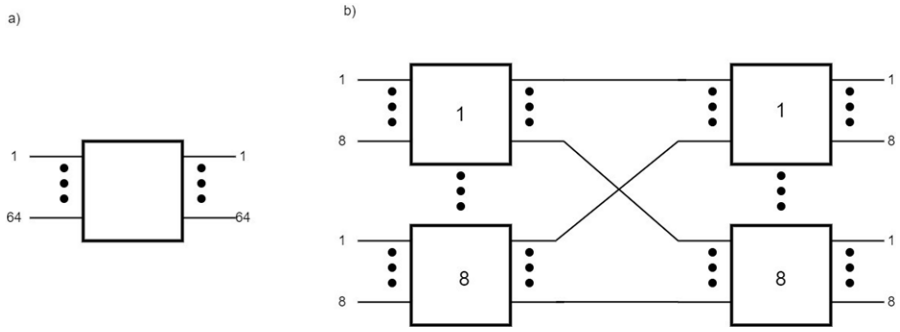

Figure 3. Commutation field with dimensions 64x64: a) single-section; b) double-section.

Another classification to consider when interconnecting switches is based on the ratio of inputs to outputs, resulting in:

- Compression networks,
- Expansion networks,
- Traffic-splitting networks.

Compression networks are those where the number of inputs exceeds the number of outputs. When the number of inputs is smaller than the number of outputs, they are referred to as expansion networks. If the number of inputs and outputs is identical, the networks are called traffic-splitting networks.

The most important classification for this work concerns the occurrence of blocking states. A "blocking state of a switching network" is defined as "a state in which a connection cannot be established between any free input-output pair, while there exist other states in the network where such a connection would be possible" [3]. Based on this definition, we distinguish the following types of networks:

- Non-blocking in the strict sense,
- Non-blocking in the wide sense,
- Rearrangeable networks,
- Repackable networks,
- Blocking networks.

A switching network is non-blocking in the strict sense if no blocking occurs, regardless of the path selection algorithm used. A network is non-blocking in the wide sense if blocking can be avoided by using an appropriate path selection algorithm. In rearrangeable networks, blocking can be resolved by modifying already established paths. In repackable networks, paths can be adjusted after other connections have ended. Networks where blocking cannot be avoided using any of the aforementioned methods are referred to as blocking networks.

Path Selection Algorithms

The literature describes numerous path selection algorithms, with the most well-known being:

- Sequential algorithm,
- Quasi-random algorithm,
- Beneš algorithm.

Sequential Algorithm

The sequential algorithm selects a path for each request by using the first free and available switch. The order of selection is predefined and fixed. This algorithm is characterized by higher loads on the initial switches, which are often occupied, requiring the algorithm to search through the switches in the middle stages.

Quasi-Random Algorithm

The quasi-random algorithm does not have a fixed starting point; it is variable. If the switch used to establish a connection has index x, the next request will be processed by switch x+1, or if that is occupied, x+2, or another switch according to the scheme x + next switch. Compared to the sequential algorithm, this one distributes the load on the switches more evenly, potentially establishing a connection in fewer attempts.

Beneš Algorithm

The Beneš algorithm selects the path for each request through the most loaded yet free switch. The load is defined as the number of connections currently handled by the switch at the time of checking. A variation of this algorithm avoids using completely free switches to establish a connection, whenever possible.

Switching Network Structures

There are various switching network structures, differing in their combinatorial properties. These include Clos networks (Figure 4), as well as banyan (Figure 5), baseline (Figure 6), and omega networks (Figure 7). The latter three networks are composed of 2x2 switches and are topologically equivalent, meaning that if a property holds for one of them, it will also hold for the others [4]. Notably, they consist of the same number of elements but differ in the way these elements are interconnected.

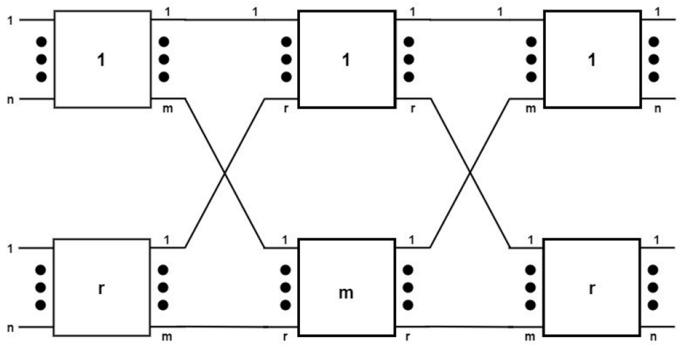

Figure 4. Three-stage Close switching fabric.

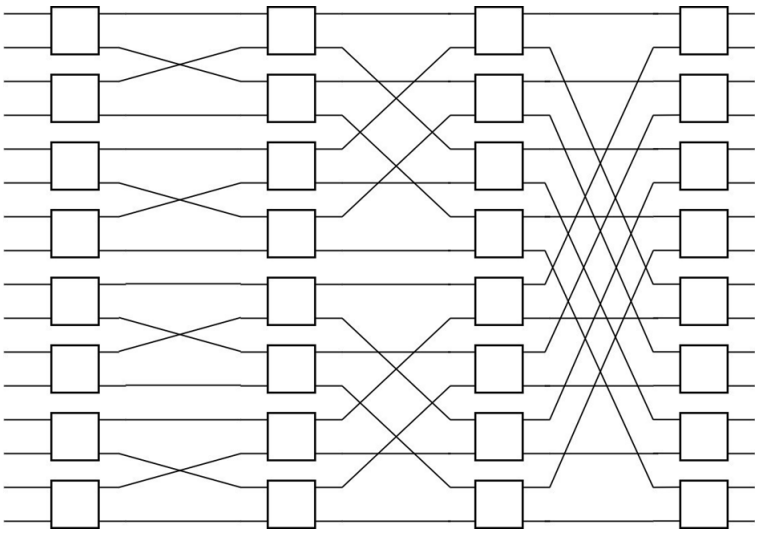

Figure 5. Banyan switching fabric.

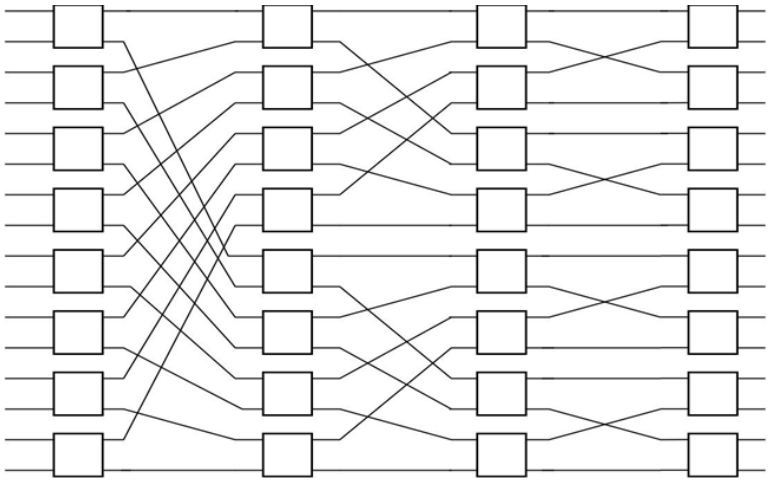

Figure 6. Baseline switching fabric.

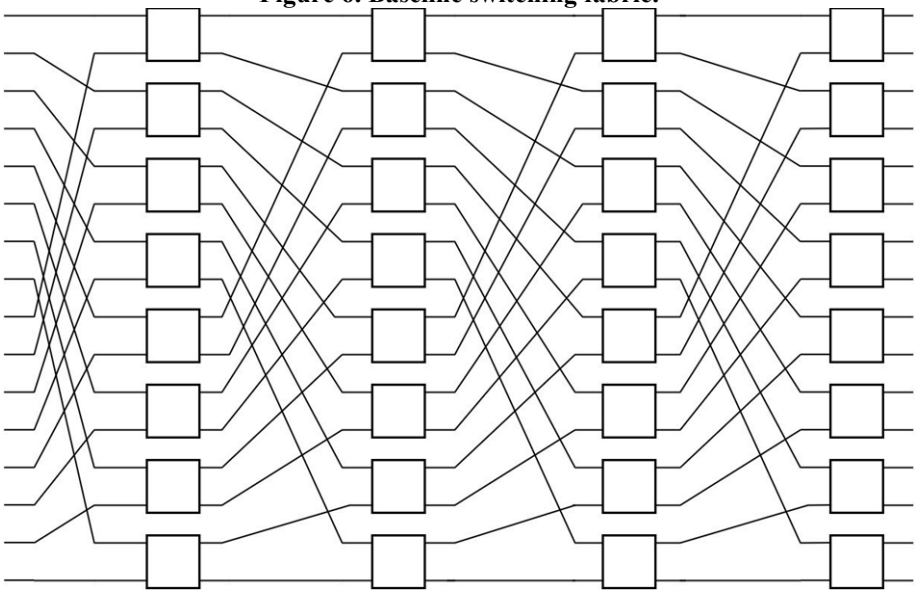

Figure 7. Omega switching fabric.

Model Preparation

Training Data

Training data is essential for training the model. In this case, it consists of data representing traffic within the switching network. The data is artificially generated using a tool provided by the supervisor. The tool accepts several input parameters, such as:

- Number of inputs for the first-stage switches and outputs for the last-stage switches (n),
- Number of switches in the middle stage (m),
- Number of switches in the edge stages (r),
- Number of load scenarios,
- Traffic intensity at the network inputs (Erl/input) for each scenario,
- Number of events,
- Name of the path selection algorithm used.

Additionally, the tool operates on a Clos network structure. Once organized, the data takes the following format: "parameter1" + separator + "parameter2" + separator, and so on.

The parameters, in order, are:

- Input number for the requested connection,
- Output number for the requested connection,
- Switch number for the first stage,
- Switch number for the third stage,
- Switch number for the middle stage.

Following this, data representing the state of the network before establishing the given connection is included in the format of a sequence of switch numbers from the first and third stages used to establish the connection, separated by the ":" character.

When setting the parameters to n=2, m=3, and r=2, the resulting switching network structure is as shown in Fig. 14.

After running the tool and extracting a sample row of the generated data, we get: 0;1;0;0;0;. In this case, a connection is established from input 0 to output 1, where switch number 0 is used in the first stage as well as in the last stage. Additionally, switch 0 in the middle stage is used to establish the connection (Figure 15).

This was one of the first rows, so there is no prior network state to read before establishing the connection.

Model Validation

Non-blocking Networks

The model was tested on non-blocking networks with a size of v(2,3,4) (Fig. 8). The training data was generated based on two algorithms: the sequential algorithm and the Beneš algorithm.

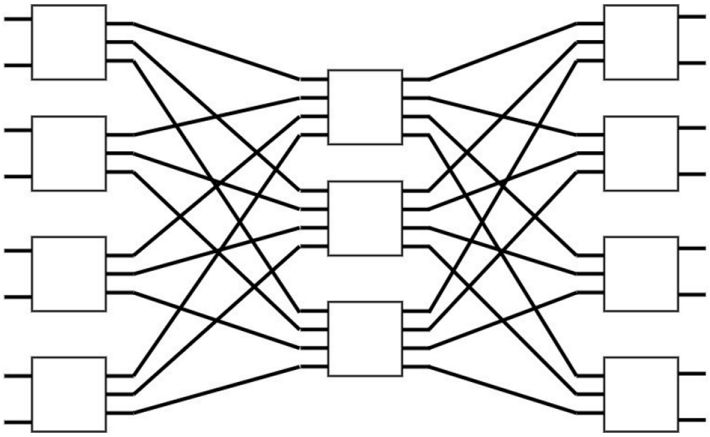

Figure 8.. v(2,3,4) switching fabric.

For both algorithms, the losses—defined as the ratio of blocked connections to the total number of requested connections—amount to zero. When the path selection algorithm was replaced by the machine learning model trained on data from the sequential algorithm, the losses for various network loads (in Erlang) are shown in Fig. 9. In the case where the model was created using data generated with the minimal index algorithm, the losses are presented in Fig. 10.

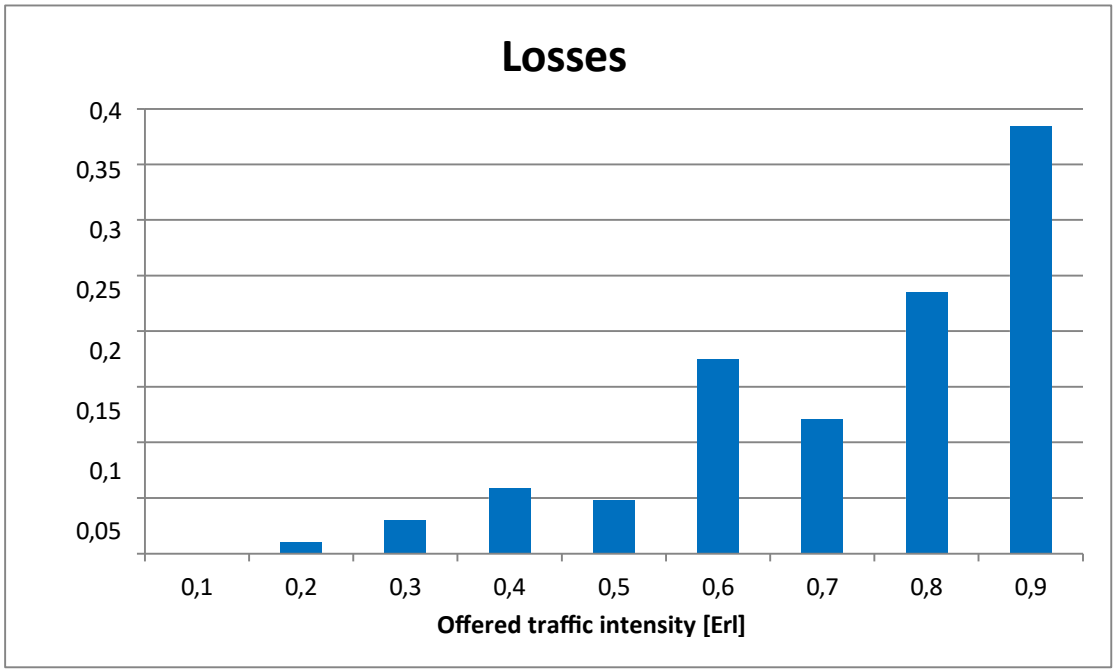

Figure 9. Losses for the model generated based on the sequential algorithm for the v(2,3,4) switching network.

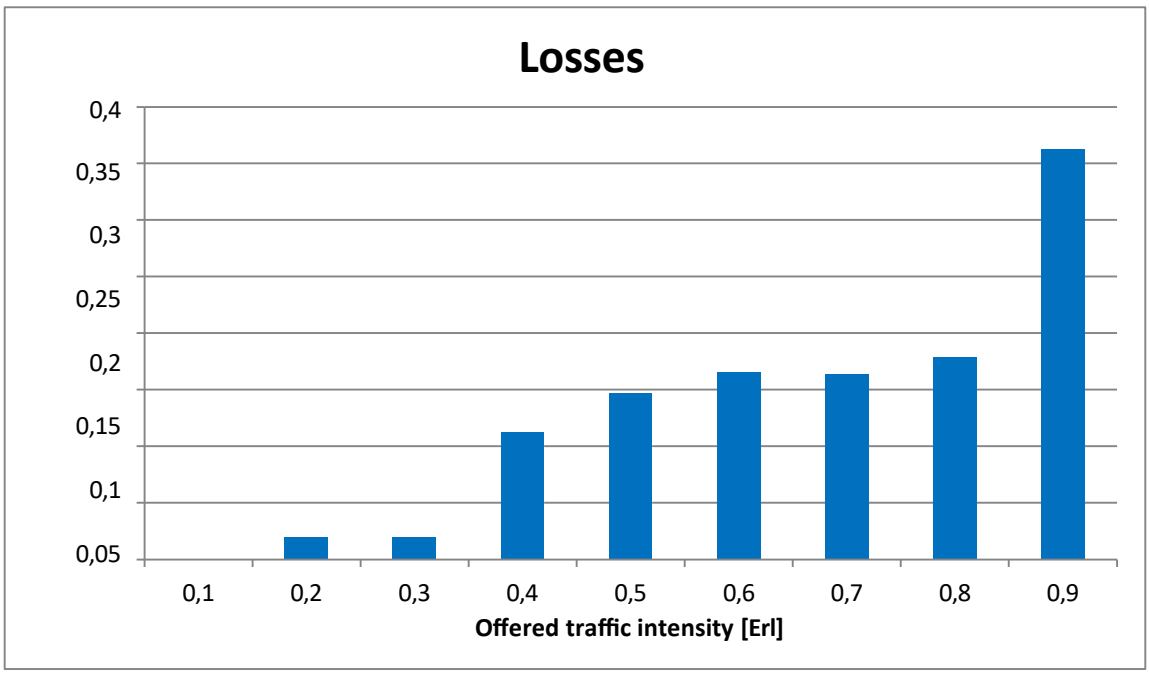

Figure 10. Losses for the model generated based on the minimal index algorithm for the v(2,3,4) switching network.

Blocking Networks

The model was also tested on blocking networks with a size of v(4,3,3) (Fig. 11). The training data was generated based on two algorithms: the sequential algorithm and the Beneš algorithm.

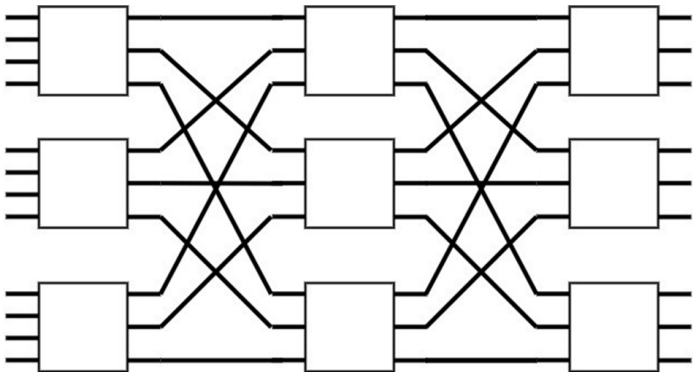

Figure. 11. v(4,3,3) switching fabri.

For the sequential algorithm and the model created using data derived from the sequential algorithm, the losses are presented in Fig. 12. Meanwhile, the losses for the minimal index algorithm and the model created using data derived from it are shown in Fig. 13.

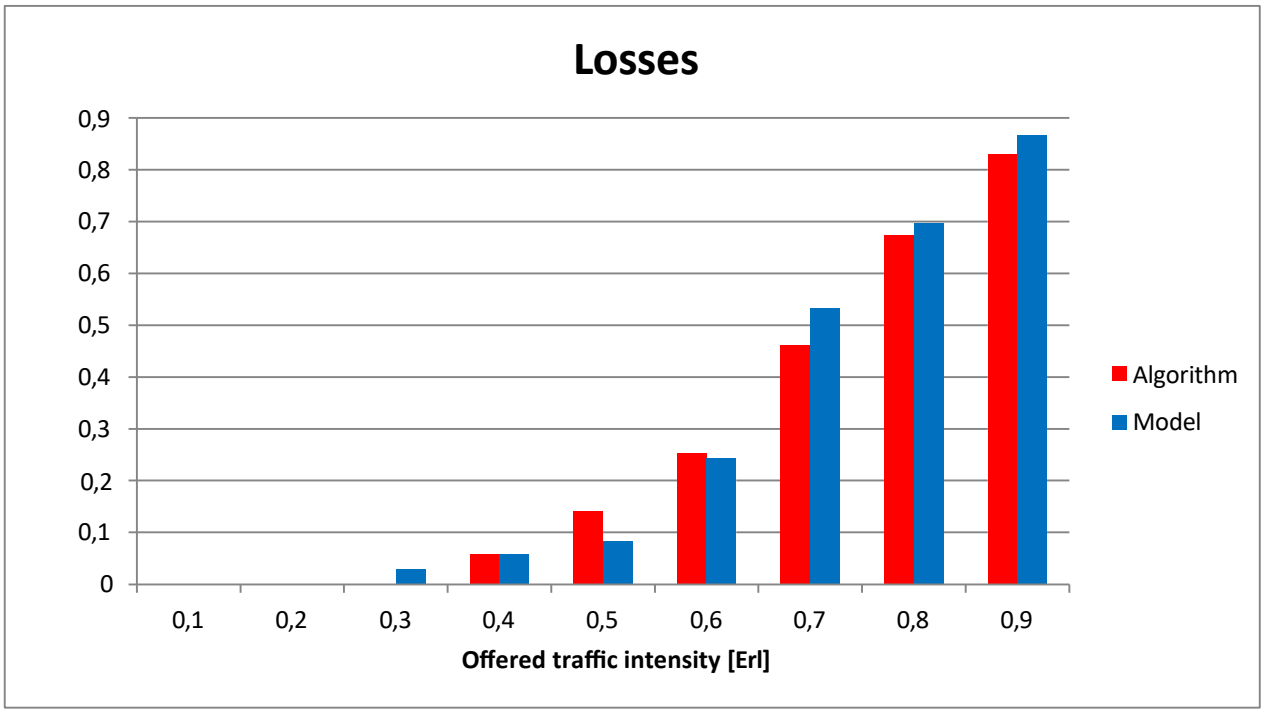

Figure 12. Losses for the sequential algorithm and model.

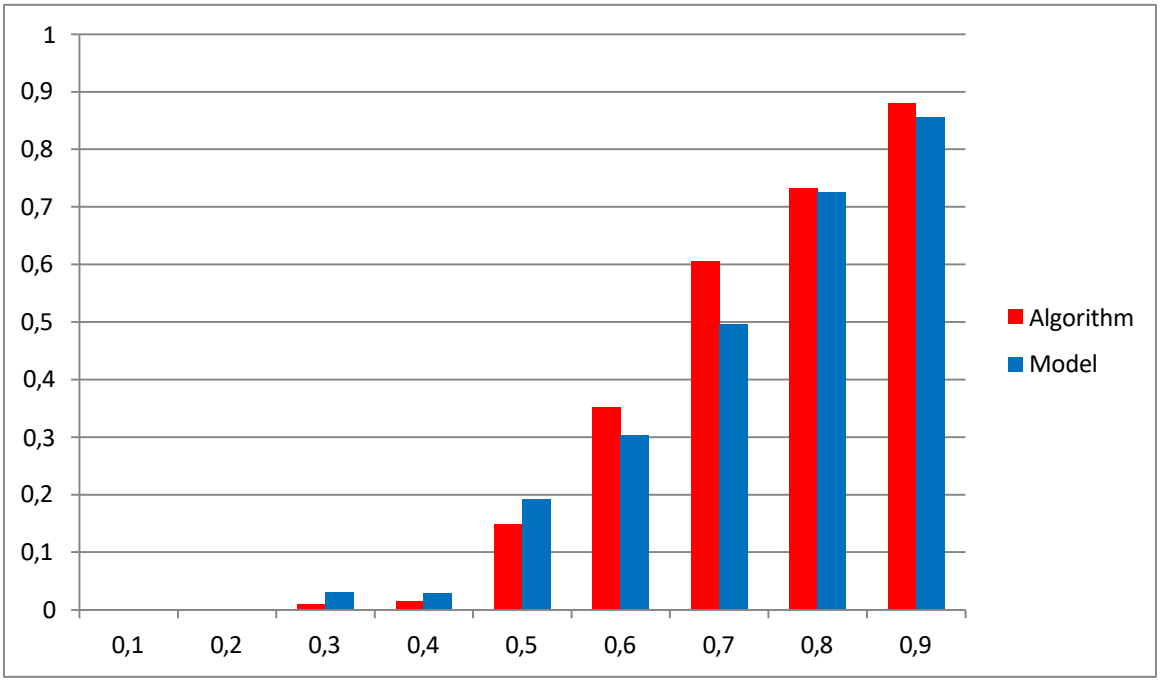

Figure 13. Losses for the minimal index algorithm and model.

Conclusions

The article presents a machine learning model based on data from path selection algorithms and conducts research to compare their performance [9].

The first model is based on data from a sequential algorithm for non-blocking Clos networks v(2,3,4). The path selection algorithm does not cause any losses because the network meets the conditions for non-blocking in the strict sense. However, the model did not perform as well. When traffic intensity does not exceed 0.5 Erlang, the results are promising, with losses not exceeding 5%. However, at higher network loads, losses reach over 30%.

The second model was based on the minimal index algorithm for non-blocking Clos networks v(2,3,4), and, as in the previous case, the path selection algorithm does not cause any losses. Similar results can be observed for the machine learning model, with the difference that the model performs worse under low network loads. Promising results are achieved only for loads up to 0.3 Erlang, while in the range of 0.4–0.8 Erlang, losses do not exceed 20% and are distributed more evenly than in the previous model.

The next presented model is designed for Clos networks that do not meet non-blocking conditions, with a size of v(3,4,4), based on the sequential algorithm. From the comparison results, it can be concluded that it produces similar losses, and in some load scenarios, even lower losses than the algorithm.

The last of the compared models is for non-blocking Clos networks v(2,3,4), created using data from the minimal index algorithm. In this case, the model proved to be slightly better than the control algorithm at higher network loads, while up to a load level of 0.5 Erlang, the algorithm was slightly more efficient.

The obtained results clearly show that the created machine learning model for non-blocking networks performs its task well only under low network loads, but unfortunately does not handle higher loads as effectively as dedicated path selection algorithms. However, when looking at blocking networks, the model performs just as well as the algorithms, and in certain scenarios, even better. This suggests that it is possible to create a model that, under specific conditions, can be a better solution than using traditional algorithms.

Acknowledgement

This research was funded by the Polish Ministry of Science and Higher Education (No. 0313/SBAD/1311).

Bibliography

- R. J. Schalkoff. *Artificial Intelligence: An Engineering Approach.* McGraw- Hill College, 2018
- R. Adib Bin, K. Ashfakul Karim, *AI revolutionizing industries worldwide: A comprehensive overview of its diverse applications*, Hybrid Advances, Volume 7, Hybrid Advances, Elsevier, 2024.
- A. Jajszczyk. Introduction to Teleinformtics. PWN 1998.
- A. Pattavina. Switching Theory Architecture and Performance in Broadband ATM Networks. John Wiley & Sons,1998.
- Matt R. Cole, *Hands-On Machine Learning with C#: Build smart, speedy, and reliable data-intensive applications using machine learning,* Packt Publishing.2018,
- Iansiti, Marco, Lakhani, Karim R. *Competing in the Age of AI : Strategy and Leadership When Algorithms and Networks Run the World*. Recorded Books: Gildan Audio, 2020
- J. STEFANOWSKI, *SVM – Support Vector Machines*, http://www.cs.put.poznan.pl/jstefanowski/ml/SVM.pdf Available at: 15.11.2024
- Nello Cristianini, John Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel based learninig methods*, Cambridge University Press, 2013
- P. Jastrzębski, Simulation Environment Of Energy-Aware Switching Fabrics, master thesis, Poznan University of Technology, Poznań, 2020