

## Article

# Improving the Energy Efficiency of Software-Defined Networks through the Prediction of Network Configurations

Manuel Jiménez-Lázaro <sup>†</sup>, Juan Luis Herrera <sup>†</sup>, Javier Berrocal <sup>†</sup>  and Jaime Galán-Jiménez <sup>\*,†</sup> 

Department of Computer Systems and Telematics Engineering, University of Extremadura, 10001 Cáceres, Spain

\* Correspondence: jaime@unex.es

† These authors contributed equally to this work.

**Abstract:** During the last years, huge efforts have been conducted to reduce the Information and Communication Technology (ICT) sector energy consumption due to its impact on the carbon footprint, in particular, the one coming from networking equipment. Although the irruption of programmable and softwarized networks has opened new perspectives to improve the energy-efficient solutions already defined for traditional IP networks, the centralized control of the Software-Defined Networking (SDN) paradigm entails an increase in the time required to compute a change in the network configuration and the corresponding actions to be carried out (e.g., installing/removing rules, putting links to sleep, etc.). In this paper, a Machine Learning solution based on Logistic Regression is proposed to predict energy-efficient network configurations in SDN. This solution does not require executing optimal or heuristic solutions at the SDN controller, which otherwise would result in higher computation times. Experimental results over a realistic network topology show that our solution is able to predict network configurations with a high feasibility (>95%), hence improving the energy savings achieved by a benchmark heuristic based on Genetic Algorithms. Moreover, the time required for computation is reduced by a factor of more than 500,000 times.

**Keywords:** SDN; machine learning; logistic regression; energy efficiency



**Citation:** Jiménez-Lázaro, M.; Herrera, J.L.; Berrocal, J.; Galán-Jiménez, J. Improving the Energy Efficiency of Software-Defined Networks through the Prediction of Network Configurations. *Electronics* **2022**, *11*, 2739. <https://doi.org/10.3390/electronics11172739>

Academic Editor: Lionel Nkenyereye

Received: 6 August 2022

Accepted: 28 August 2022

Published: 31 August 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The energy consumption problem in communication networks was one of the most studied problems in the networking area during the 2010–2020 decade due to its negative impact on the environment. The implications of the power consumption generated by the Information and Communication Technology (ICT) sector, and especially by the networking equipment, led researchers in the area to prioritize their efforts to reduce the carbon footprint [1,2].

In recent years, sustainability has gained importance worldwide. Several studies have been published showing the positive impact that ICT solutions may have on sustainability [3,4]. In addition, during the COVID-19 pandemic, it has been demonstrated that ICT solutions, such as remote working, video meetings, etc., have been instrumental in keeping businesses and societies going, thereby proving these solutions for real. Moreover, with the advent of 5G and 6G technologies, in which the goal is to pursue a fully connected intelligent world, efforts must be made to keep alleviating the impact of the emissions generated by the telco sector.

Focusing on the networking area, a first concern is related to the inefficiency of network devices: (i) they are always active regardless of their use and (ii) their power consumption is independent from their traffic load. Moreover, networks are usually designed to avoid congestion during peak traffic periods. In order to reduce the network power consumption, *green networking* [2] aims at making the network consumption load-dependent. Different techniques have been developed in recent years: (i) the use of re-engineering approaches with more energy-efficient technologies such as Energy Efficient

Ethernet—IEEE 802.3az [5]; (ii) the exploitation of dynamic adaptation approaches, in which the modulation of capacities is considered according to traffic load [6]; and (iii) the use of sleeping methods, in which the goal is to use the least number of active devices (links, routers) so that the highest power savings are experienced [7].

The increasing interest in the application of Artificial Intelligence (AI) and Machine Learning (ML) techniques to the networking area [8,9] opens a new niche that accounts for the improvement of the energy efficiency of communication networks. If the traditional energy efficient approaches [1,10,11] are combined with the use of AI/ML techniques, a step further in the improvement of energy efficiency in communication networks will be achieved. Moreover, the central vision, flexibility and programmability of the Software-Defined Networking (SDN) paradigm enable this combination of techniques to re-think and build highly energy-efficient programmable networks [12–14]. Through the implementation of ML algorithms to be run at the SDN controller, control actions in the network may be sped up to modify the network configuration according to the traffic load with the final goal of minimizing the network power consumption [15].

In this work, we investigate how to use ML to target the *H Consumption Problem* (PCP) in the context of computer networks. The main goal of the PCP is to find a feasible network configuration for a given Traffic Matrix (TM) where the number of active network devices is minimal to save the highest amount of energy. In small or medium-sized network topologies, PCP is formulated and solved using Integer Linear Programming (ILP), which outputs an optimal network configuration in terms of energy consumption [16]. However, the PCP is known to be NP-hard, and it is not tractable to find an optimal solution for large networks [16]. In this way, heuristic algorithms are required to obtain (near) optimal solutions in a short period of time. Specifically, we transform the PCP problem into an ML-based classification problem. To that end, the solution to the PCP is provided by a classifier in response to a given TM passed as input.

A Logistic Regression-based Energy Efficient (LR-EE) algorithm is proposed and trained with historical data from a big dataset of TMs. Then, the classifier is able to provide a (near) optimal network configuration for a new (and probably unseen) TM without the need for running the ILP or a heuristic, thus speeding up the process in an online manner. Traffic load thresholds exploited by the SDN controller are considered to run the LR-based ML algorithm in order to potentially change the network configuration for an increase in the energy that is saved.

The main contributions of this work are:

- The definition of a novel algorithm to predict energy-efficient network configurations based on Logistic Regression (LR).
- The evaluation of the proposed LR-based algorithm over a realistic network topology.
- The comparison of the obtained results with energy-efficient ad hoc solutions.

The rest of the paper is organized as follows. Section 2 introduces some related work. A review on the PCP and the heuristic considered to solve it are described in Section 3. Section 4 describes the system model. Section 5 includes the description of the proposed ML algorithm. Experimental results are reported and analyzed in Section 6. Finally, Section 7 draws some conclusions and future works.

## 2. Research Gap

Over the last few years, extensive work has been conducted on power consumption management to improve the energy efficiency of communication networks. For instance, some works, such as [17–20], focus on reducing the energy consumption of satellite and terrestrial networks while trying to maximize the quality of service. Recent works also focus on SDN networks. The flexibility that these networks provide, such as the separation of the data plane from the control plane and its consequent advantages, creates new opportunities to define more dynamic and energy-aware networks.

Recent works in this area have focused on ILP-based approaches [21,22]. These works allow researchers to identify the optimal network configuration, but they cannot be used

for large networks due to the complexity of the problem to address. Since the power consumption problem in SDN can be modeled as a Multi-Commodity Network Flow problem, which is known to be NP-hard, other works focused on applying heuristics, such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO), among others [11,22]. Nevertheless, compared to the dynamicity that SDN networks (time scale of the order of milliseconds) require, these techniques take longer to identify an appropriate network configuration.

In order to reduce the re-configuration time, ML is used to improve network performance. For example, in the field of Traffic Engineering (TE), several works exploit the use of Reinforcement Learning to adapt the network configuration to the current traffic load and minimize the Maximum Link Utilization (MLU) [23,24]. In order to save energy, Ref. [25] proposes an algorithm based on DRL to predict traffic in order to optimize the energy efficiency and perform real-time load balancing. To the best of our knowledge, this is the closest solution to the problem we aim to tackle. In this paper, we focus on using Logistic Regression with the aim of reducing the complexity of the problem addressed. This technique is used to identify the subset of links to be turned off depending on the network traffic load.

In the following section, the power consumption problem is studied in order to achieve the goal of this article, bearing in mind the previous works that have just been presented.

### 3. A Review on the Power Consumption Problem

In this section, a review on the power consumption problem in computer networks is provided. Both the classical ON-OFF model [26] and the one based on the Adaptive Link Rate technique [6,11] are analyzed. Then, a heuristic-based solution is proposed to solve the PCP in tractable time. This heuristic will be used to generate data to feed the proposed ML algorithm for the classification of TMs and the assessment of their corresponding network configuration.

#### 3.1. The Power Consumption Problem (PCP)

PCP is typically modeled as a Multi-Commodity Flow (MCF) problem [16] in which the network is represented by a network graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , with  $\mathcal{V}$  as the set of nodes and  $\mathcal{E}$  as the set of unidirectional links connecting them. Each link  $l_{i,j} \in \mathcal{E}$  has a specific capacity of  $C_{i,j}$  units to accommodate traffic flows and a power consumption of  $p_{i,j}$  Watts if they are active and the classical ON-OFF power consumption model is adopted [26]. In case the Adaptive Link Rate (ALR) approach is considered [6,11], a link  $l_{i,j} \in \mathcal{E}$  operating at rate  $k$  has a power consumption of  $p_{i,j}^k$  Watts;  $p_{i,j} = 0$  in case the link is powered off (sleeping). The MCF-PCP requires a traffic matrix  $\mathcal{T}$  as input, with the description of the volume of traffic per source-destination flow  $f_{i,j}$ , represented by  $d_{i,j} \in \mathcal{T}$ . Considering the network features described above, the PCP aims at finding a network configuration  $\mathcal{G}' \subseteq \mathcal{G}$  with the minimum power consumption that respects: (i) link capacity constraints, i.e., the traffic load of each link must be  $C_{i,j} \leq 100\%$ ; and (ii) flow conservation constraints, i.e., the amount of traffic reaching a node must be equal to the volume of traffic leaving such node, excluding the traffic inserted/terminated at that node.

More formally, the Integer Linear Programming (ILP) formulation intended to solve the PCP requires a set of variables that are described as follows (The problem described refers to the classical link switch off problem. In case ALR is adopted, variable  $x_{i,j}^k$  must be considered, along with an additional constraint to limit the maximum number of rates,  $k$ , per link.):

- $x_{i,j}$  is a binary variable whose value is equal to 1 if the link  $l_{i,j}$  is active; 0 if the link is powered off.
- $f_{i,j}^{s,d}$  is a binary variable whose value is equal to 1 if the traffic demand of volume  $d_{s,d}$  derived by flow  $f_{s,d}$  is routed on the link  $l_{i,j}$ ; 0 otherwise.

After defining the required variables, the PCP formulation is described by Equations (1)–(3):

$$\min \sum_{l_{i,j} \in \mathcal{E}} x_{i,j} p_{i,j} \tag{1}$$

subject to:

$$\sum_{j \in \mathcal{V}_i^-} f_{i,j}^{s,d} - \sum_{j \in \mathcal{V}_i^+} f_{j,i}^{s,d} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = d \\ 0 & \text{if } i \neq s, d \end{cases} \quad \forall i \in \mathcal{V}, d_{i,j} \in \mathcal{T} \tag{2}$$

$$\sum_{d_{s,d} \in \mathcal{T}} f_{i,j}^{s,d} \cdot d_{s,d} \leq x_{i,j} \cdot C_{i,j} \quad \forall l_{i,j} \in \mathcal{E} \tag{3}$$

Equation (1) aims at minimizing the network power consumption by finding a suitable network configuration that minimizes the number of active links subject to the constraints defined in Equations (2) and (3). Equation (2) describes the classical flow conservation constraints. It imposes that the volume of traffic that reaches a node must be equal to the amount of traffic that passes through it toward the next hop, unless the node is a source or a destination node. Equation (3) represents the link capacity constraint, where the amount of traffic on the link must be, at most, the capacity of the proper link,  $C_{i,j}$ . Since the ILP formulation falls into the category of MCF problems, which are known as NP-hard problems [16], heuristic solutions are required to validate their benefits over topologies of large size, and in tractable times. In this paper, the Genetic Algorithm (GA)-based solution proposed in [22] is used to find (near) optimal network configurations in terms of power consumption. Then, the output of the GA will be the inputs for our proposed LR-based ML algorithm.

### 3.2. GA-Based Heuristic for Power Consumption Minimization

In order to find a network configuration that minimizes the network power consumption and satisfies a given TM respecting flow conservation and link capacity constraints, we rely on the use of our GA-based solution [22]. This solution outputs a network configuration close to the optimal in terms of energy savings. In the following, we review the main aspects of the considered approach detailing the definition of the individuals that compose the population, the fitness function, and the considered biological operators (selection, crossover and mutation).

#### 3.2.1. Chromosome Definition

GAs require a population of individuals, namely chromosomes, that represent potential solutions to the PCP. In our case, a chromosome  $c \in \mathcal{P}$  represents a potential network configuration as a succession of  $L$  genes, where the  $k$ -th gene,  $g_k \in c$ , describes the operational mode  $x_{i,j}$  of link  $k = l_{i,j} \in \mathcal{E}$ :

$$c = \{g_{1,2}; g_{1,3}; \dots; g_{i,j}; \dots; g_{N,N-1}\} \tag{4}$$

Thus, for the classical ON-OFF power consumption model, binary variables are considered, i.e.,  $g_{i,j} = 1$  in case link  $l_{i,j}$  is active;  $g_{i,j} = 0$  otherwise. If the ALR model is adopted, then  $K$  values are considered for each link configuration, i.e.,  $g_{i,j} = s, s \in [0, K - 1]$ .

### 3.2.2. Fitness Function

A fitness function is also required by GAs to evaluate the goodness of each chromosome (solution) of the population. Related to the objective function defined in Equation (1), Equation (5) is applied to each individual in the population. Function  $P(c)$  assesses the aggregated power consumption of the network configuration represented by the potential solution  $c$ , according to the current operational mode  $p_k$  of each link  $k$  in the network. The resulting value of the sum is multiplied by  $\theta$ , which is set to 1 if the network configuration mapped by  $c$  is feasible, i.e., the TM can be correctly routed according to the shortest path rule without violating any of the constraints reported in Equations (2) and (3). Otherwise,  $\theta$  takes a value high enough to penalize the corresponding fitness value to such an unfeasible chromosome.

$$P(c) = \left( \sum_{g_k \in c} g_k \cdot p_k \right) \theta, \quad \forall k \in \mathcal{E} \quad (5)$$

### 3.2.3. Biological Operators

In order to perform the evolution procedure, GAs apply biologically inspired operators (selection, crossover, mutation) to the individuals in the population. The set of individuals that survive and form part of the next generation in the GA is selected by applying the classical roulette wheel criterion. Moreover, the combination of individuals to generate offspring is performed by means of the single-point cross-over function. Regarding the mutation process, a two-step uniform mutation function is applied. First, a fraction of each individual is selected for mutation. Every gene in this fraction has a probability rate of being mutated. The second step is to replace each selected gene by another valid value. The application of selection, crossover and mutation operators is repeated in each generation of the GA-based algorithm.

## 4. System Model

As introduced in Section 1, the main goal of this work is to define a model that is able to add energy efficiency features to SDN networks by putting unneeded links to sleep depending on the current traffic load. To achieve this goal, a Logistic Regression-based ML algorithm has been used.

The proposed algorithm is implemented in SDN networks, which are composed of SDN switches that represent the data plane of the network. Moreover, there is a centralized element that has a global view of the network and is in charge of determining the routing logic. This centralized element is the SDN controller and has a global view of the network, composing the control plane. The SDN controller is able to monitor the network elements by exchanging messages with them. These messages are defined by the OpenFlow protocol [27]. They allow the SDN controller to assess the traffic load as well as network statistics to obtain network metrics such as MLU, average delay or packet loss.

Figure 1 represents the whole process that is carried out for the proposed ML algorithm to work in an SDN environment. As can be seen, a 6-node network with 10 links is represented. First ①, the SDN controller estimates a TM based on average data from historical measurements, as in [28]. After having the TM calculated, the ML algorithm (which is installed as an application inside the controller) is executed by passing the TM as input. The ML algorithm, depending on the learning process of its model, will assign a specific configuration to the network, which implies turning off specific links and keeping the rest active. In the case of Figure 1, it can be seen in ② that the output of the ML algorithm application determines whether to turn off 4 out of the 10 links, thus saving 40% energy and still being able to satisfy the estimated traffic demand.

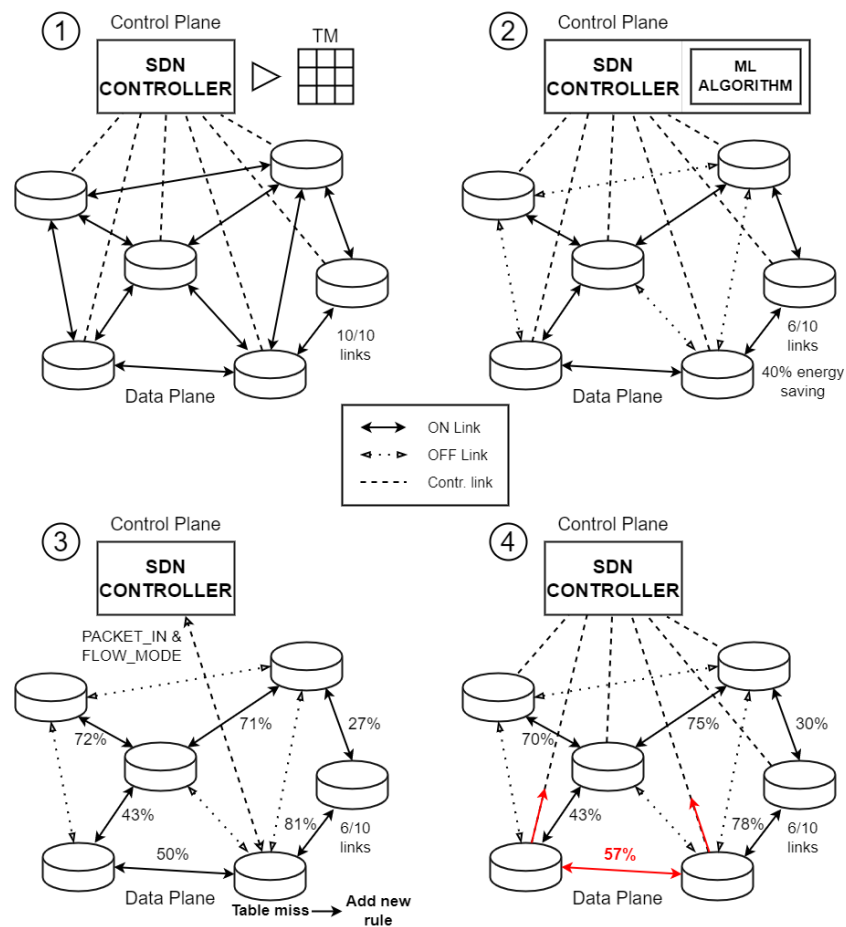


Figure 1. System model overview.

The flow table of each SDN switch stores rules with routing information to carry packets from their source to their destinations. The dynamicity of the network traffic implies that such rules are constantly changing. The routing information will be modified depending on the packets being routed at any given time and the network configuration with the set of active and powered off links. If a flow table does not store the rules that allow a packet intended to be sent to reach its destination, the SDN switch will inform the controller of this situation by sending a PACKET\_IN message, as can be seen in ③. With this message, the node is requesting the controller to add, modify or delete a rule in order to route the incoming packet. Thus, the controller replies with a FLOW\_MODE message. After each execution of the ML algorithm, the network configuration may change, so new rules must be added, removed or modified, to re-route the packets.

Therefore, since the network configuration is applied after the execution of the ML algorithm, the traffic load of each link may change. Thus, the controller asks the switches about the load of their links from time to time by means of the OFPIT\_STAT\_TRIGGER message [27]. In our model, it is defined that if the traffic load of a link differs by 5% since the previous configuration was applied, the switches that are connected by the link will notify back to the SDN controller. As can be seen in ④, the load of the red link passed from 50% to 57%, thus triggering the process to notify the controller. At this point, as the controller knows the network traffic load, it is able to assess a new TM, going back to ②, where the ML algorithm is called again with the new TM. Indeed, the ML algorithm will determine the new configuration that fits the new network situation and the process is iteratively repeated.

## 5. Logistic Regression-Based Energy Efficient Algorithm

In this section, the algorithm that is proposed to predict energy-efficient network configurations based on Logistic Regression (LR-EE) is described. Thus, a methodology to adapt the PCP into a *supervised classification problem* is provided. The aim is to apply a solution based on LR to predict the network configuration associated to a TM and save energy. However, before applying the supervised ML algorithm based on LR, a method for the reduction of the number of classes must be considered. In this sense, a non-supervised ML algorithm is first considered for such a reduction.

### 5.1. Clustering Process for Network Configurations Reduction

As described in Section 3, each TM is taken as input by the GA results in an associated network configuration. These network configurations indicate which links are active and which ones are put to sleep to save energy, along with a suitable routing for the given TM. In the worst-case scenario, the application of the GA over a set of different TMs may report a set of network configurations that are all different. In the case of thousands of different TMs, thousands of network configurations can be assessed, with an upper bound of  $2^{\mathcal{E}}$  due to the binary values of the genes. On the other hand, it may happen that the same network configuration can be applied to a subset of TMs.

Since the main goal of the ML algorithm proposed in this work is to perform a classification of TMs and assign them a valid network configuration able to route the traffic and save energy, a big dataset is required for the training process. However, a first problem related to the number of classes must be tackled. If there are many different network configurations for the prediction, the ML algorithm may not learn correctly due to the ratio between traffic matrices and configurations to be classified. Therefore, its performance may be low.

In order to reduce the number of classes, a non-supervised ML algorithm based on the concept of *K-means* [29] is first applied to group the set of input TMs to  $\phi$  different clusters. The process followed by the algorithm is as follows: each TM belonging to the data set is mapped to one of the  $\phi$  network configurations representing the different clusters. For each cluster, there is only a single network configuration that is valid for all the TMs belonging to the cluster. In this way, a dimensionality reduction is performed:

- If none of the original configurations are valid for all the TMs belonging to that cluster, the configuration with the highest number of active links is selected, and links are iteratively switched on until a valid configuration for all TMs is found.
- If there is an original configuration that is valid for all the TMs in that cluster, it is selected.
- If there is more than one original configuration that is valid for all the TMs in that cluster, the one with the highest number of links off is selected (highest energy savings).

With this approach, we assume a potential gap in terms of energy savings. However, a reduced number of classes will lead to a better classification with the application of the proposed LR-based ML algorithm described next.

### 5.2. Turning the PCP into a Supervised Classification Problem

In the following, a description of the methodology followed to convert the PCP into a supervised classification problem is provided.

Let us denote  $d_{i,j} \in \mathcal{T}$  as the volume of traffic to be sent from node  $i \in \mathcal{V}$  to node  $j \in \mathcal{V}$ , with  $i \neq j$ . The GA-based solution reports as output a (near) optimal network configuration  $C_{\mathcal{T}} = (\mathcal{S}_{\mathcal{T}}, \mathcal{R}_{\mathcal{T}})$  with the status of each link in the network (active or powered off),  $\mathcal{S}_{\mathcal{T}} = \{p_{i,j}\}, \forall i,j \in \mathcal{E}$ , and the routing configuration  $\mathcal{R}_{\mathcal{T}}$  for each source-destination flow. As an example, let us consider the topology shown in Figure 2a, which is composed of six nodes connected by eight links. One potential solution of the GA is the one depicted

by Figure 2b, in which the optimal configuration to route a specific TM  $\mathcal{T}_1$  saving the most amount of energy is  $C_{\mathcal{T}_1} = (\mathcal{S}_{\mathcal{T}_1}, \mathcal{R}_{\mathcal{T}_1})$ . Link configuration is given by Equation (6):

$$\begin{aligned} \mathcal{S}_{\mathcal{T}_1} &= \{s_{1,2}; s_{1,3}; s_{2,4}; s_{2,5}; s_{3,4}; s_{3,5}; s_{4,6}; s_{5,6}\} \\ &= \{0; 1; 1; 0; 1; 1; 1; 0\} \end{aligned} \tag{6}$$

where five links are active ( $s_{1,3}; s_{2,4}; s_{3,4}; s_{3,5}; s_{4,6}$ ) and three links are powered to save energy ( $s_{1,2}; s_{2,5}; s_{5,6}$ ). This leads to 37.5% power savings. The associated routing configuration for all the flows originated by node 1 is shown in Figure 2c. As it can be seen, each flow avoids using the powered off links, which can result in an increase of the traffic volume on specific links such as, e.g.,  $l_{3,4}$ . Thus, the routing configuration is given by Equation (7):

$$\begin{aligned} d_{1,2} &\longrightarrow (l_{1,3}, l_{3,4}, l_{4,2}) \\ d_{1,3} &\longrightarrow (l_{1,3}) \\ d_{1,4} &\longrightarrow (l_{1,3}, l_{3,4}) \\ d_{1,5} &\longrightarrow (l_{1,3}, l_{3,5}) \\ d_{1,6} &\longrightarrow (l_{1,3}, l_{3,4}, l_{4,6}) \\ &\vdots \\ d_{i,j} &\longrightarrow h_{i,j} \\ &\vdots \\ d_{6,5} &\longrightarrow (l_{4,6}, l_{3,4}, l_{3,5}) \end{aligned} \tag{7}$$

where the path for traffic demand  $d_{i,j} \in \mathcal{T}_1$  according to routing configuration  $\mathcal{R}_{\mathcal{T}_1}$  is  $h_{i,j}$ . For instance, the flow originated at node one destined to node six must traverse links  $l_{1,3}$ ,  $l_{3,4}$  and  $l_{4,6}$ . Thus, the supervised classification problem receives as input the serialized TM  $\mathcal{T}_1$  with the set of demands to be routed and provides as output a near optimal network configuration  $C_{\mathcal{T}_1}$ . Generally, the execution of the GA outputs a specific network configuration. In the worst-case scenario, this can result in a situation in which each TM is associated with a different network configuration. However, in practical scenarios, many of the configurations can be applied to a set of the considered TMs (and not only to one TM), thus reducing the space of configurations under consideration. Therefore, if the number of network configurations to classify the TMs with is reduced, the complexity of the classification problem is lower.

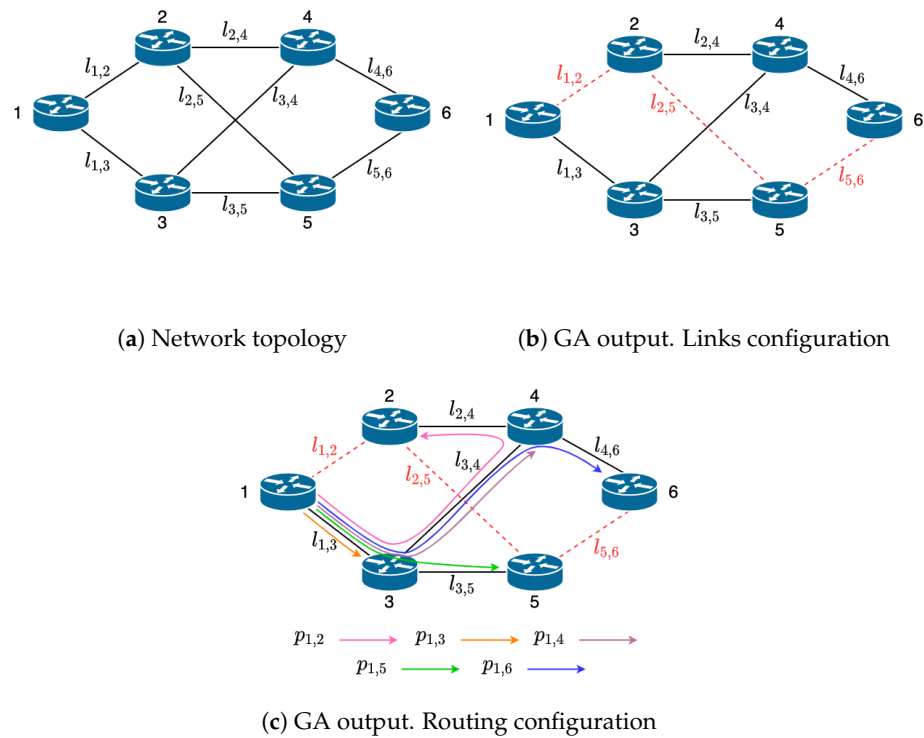
In order to turn the PCP into a supervised classification problem, we consider the input variables as the elements of a serialized TM, with a tuple  $t_k$  of type:

$$t_k = \{d_{1,2}, d_{1,3}, \dots, d_{i,j}, \dots, d_{N,N-1}\}, \forall i, j \in \mathcal{V} \tag{8}$$

and the output labels are the associated network configurations  $C_{\mathcal{T}}$  to such tuple  $t_k$  (TM) obtained by the GA. This dataset can then be used to train a classical supervised ML algorithm based on Logistic Regression. With a big enough number of tuples  $(t_k, C_{\mathcal{T}})$ , the trained ML algorithm would be able to predict a network configuration for a given TM without the need of executing the GA-based solution. Moreover, for a new TM not considered in the training set, the ML model should be able to generalize and produce a valid network configuration able to save as much energy as possible close to the one that would be obtained by the GA.

Then, in order to classify new and potentially unseen TMs, the RL-based ML algorithm is invoked and returns a specific network configuration that is expected to (i) be valid, and (ii) able to save energy. With the application of such an algorithm by the SDN controller, the GA is no longer needed and the computation time will be reduced.





**Figure 2.** Example of GA output on a 6-node topology. (a) Network topology. (b) GA output. Links configuration. (c) GA output. Routing configuration.

### 6. Experimental Results

In this section, an experimental evaluation of the proposed solution is provided. At first, the simulation environment is described. Next, a performance evaluation of the proposed solution in terms of ML and network metrics is carried out to analyze its benefits and potential drawbacks, along with the energy savings achieved.

#### 6.1. Simulation Set-Up

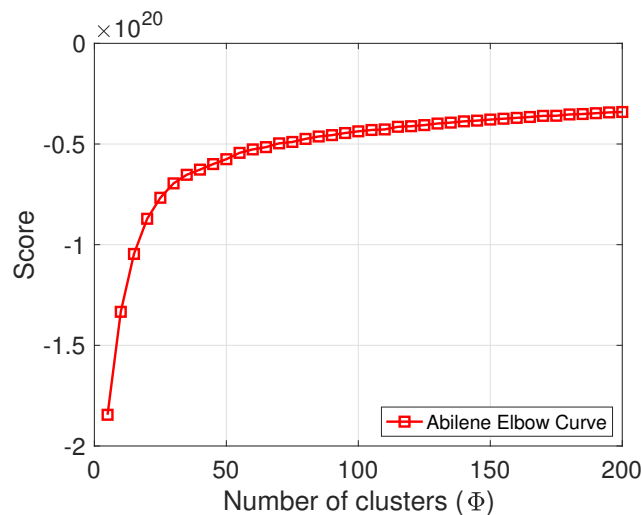
In order to evaluate the effectiveness of the proposed ML solution, Abilene topology (12 nodes and 30 links) is considered. A set of 3871 TMs retrieved from [30] are taken as input, with a time granularity of 5 minutes. Link capacities are set as follows. First, we select the peak TM in the dataset and route it over a set of shortest paths derived after the application of the Dijkstra algorithm on the network graph. After this step, each link  $l_{i,j}$  is carrying an amount of traffic  $t_{i,j}$ . Then, we assume that the capacity of each link can be upgraded by installing a set of line cards. A line card has a capacity of  $\Delta_C$  equal to  $0.5 \max_{l_{i,j}}(t_{i,j})$ , i.e., the half of the traffic carried by the link with highest link utilization.

Finally, we consider installing the minimum number of line cards needed by each link to make their utilization not greater than 100%, i.e.,  $MLU = 1$ .

Regarding the power consumption model that is considered, we assume that links can be either powered off (sleeping) or powered on (active). All the links in the network have the same power consumption when they are active ( $p_{i,j} = 1$ ). It is worth remarking that, although this assumption is unrealistic (there can be different types of links with different values of power consumption [11]), it is a classical approach followed in a big set of works tackling the power consumption problem in computer networks [7,26,31].

## 6.2. Performance Evaluation

As previously introduced, the GA-based solution outputs a (near) optimal network configuration for each TM passed as input. Then, in case Abilene topology is considered, 3871 network configurations are assessed, one per TM. In order to find a suitable number of clusters to group all the TMs, a prior analysis has been carried out. Figure 3 shows the score after the application of the K-means algorithm as a function of the number of clusters. The score is the average of the inverses of the distances of each configuration from its centroid configuration. Clearly, the score is reduced with the number of clusters, following a logarithmic pattern. In particular, a sharp increase is experienced in the range  $\phi = [1, 50]$ , and the line is stabilized with a large number of clusters. In order to evaluate the impact of the cluster size on the effectiveness of the proposed solution, four values of  $\phi$  are considered in the following performance analyses:  $\phi = \{10, 50, 100, 200\}$ . Furthermore, in order to carry out the training and testing of LR-EE, the set of TMs has been divided into two subsets. One for training (66% of the data), and the other for evaluating the performance of the algorithm (33% of the data).



**Figure 3.** Score of K-means vs. number of clusters ( $\phi$ ).

Tables 1 and 2 report the results obtained by LR-EE for the different values of  $\phi$  for comparison with the GA-based solution. Two different types of metrics are shown: (i) ML metrics, with precision, recall and F1-score for each model (Table 1); and (ii) network metrics such as the average link load, MLU, average number of hops per flow, maximum number of hops per flow, average energy saving gap (which is assessed as the energy savings of the original configuration outputted by the GA minus the energy savings of the network configuration of the cluster) and the feasibility, which is the percentage of predictions that, whether or not they are correctly predicted, are valid for the associated TM (Table 2).

**Table 1.** Machine Learning metrics for LR-EE and GA.

	ML Metrics			Computation Time	
	Precision	Recall	F1-Score	Train. T	Exec. T
LR-EE $\phi = 10$	0.84	0.84	0.84	7.67 s	1.5 $\mu$ s
LR-EE $\phi = 50$	0.76	0.76	0.75	3.10 s	1.9 $\mu$ s
LR-EE $\phi = 100$	0.75	0.74	0.73	3.20 s	2.2 $\mu$ s
LR-EE $\phi = 200$	0.74	0.72	0.71	5.41 s	4.2 $\mu$ s
GA	-	-	-	-	2.21 s

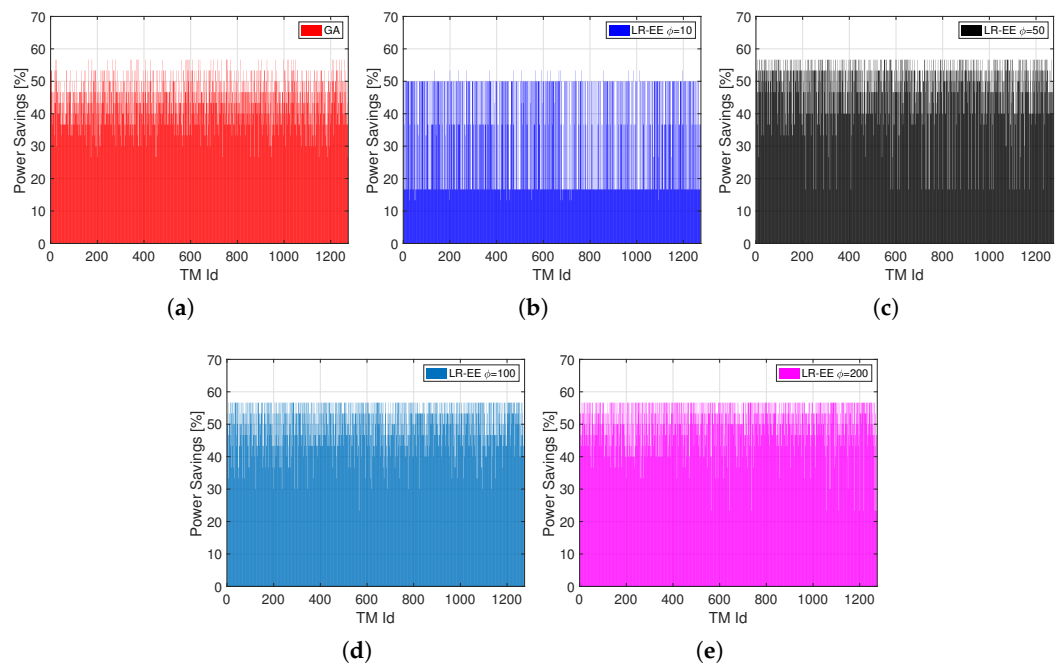
**Table 2.** Network metrics for LR-EE and GA.

	Network Metrics					Feasibility
	max_LL	avg_LL	avg_hops	max_hops	avg_gap	
LR-EE $\phi = 10$	0.26	0.99	3.60	10	11.25%	97.25%
LR-EE $\phi = 50$	0.42	1	4.50	11	−3.90%	97.80%
LR-EE $\phi = 100$	0.45	1	4.65	11	−5.96%	97.10%
LR-EE $\phi = 200$	0.47	1	4.76	11	−6.95%	95.53%
GA [22]	0.45	1	4.14	11	-	100%

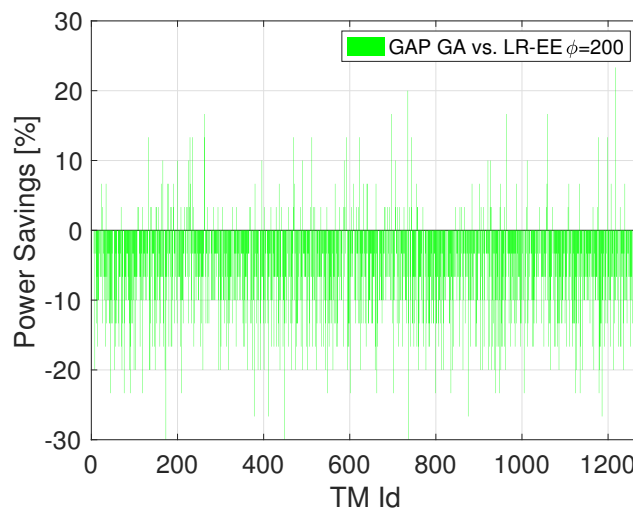
Results show that as the number of clusters ( $\phi$ ) increases, the ML metrics perform worse. This is due to the fact that if there is a high number of classes to associate the TMs with, it will be more difficult for the LR algorithm to determine to which class a TM belongs. This is partly because there are classes that have very few associated TMs, while other classes have hundreds of associated matrices. However, this is not reflected in the network metrics. The higher the number of clusters, the better the average energy saving gap is. This is because the configuration associated with each cluster can be more specific to the associated TMs, resulting in a higher number of powered off links. As a result, links are more loaded and flows require a higher number of hops to reach their destination. In fact, for  $\phi = 10$ , ML prediction is 10% better compared to the case of  $\phi = 200$ . However, worse outcomes in the energy savings are obtained for a small number of clusters, e.g.,  $\phi = 10$ , where an average gap of 11.25% compared to the GA is obtained. On the contrary, significantly better results are obtained for the case of  $\phi = 200$ , where 6.59% of energy is saved, on average, when it is compared with the GA. Moreover, the feasibility remains stable above 95% for all tests. It is therefore appropriate to select the option that saves the highest amount of energy, i.e.,  $\phi = 200$ . Finally, it can be seen that the reduction in execution time of the ML-based solution compared to GA is notable. The LR-EE configuration prediction time has an acceleration factor between 526,190 and 1,473,333 times higher than the time needed by GA to generate a new configuration.

Figure 4 reports the power savings achieved by the GA (Figure 4a) and by LR-EE (Figure 4b–e) for different values of  $\phi$  as a function of the TM Id. Note that TMs are sorted according to their traffic load in ascending order. Clearly, it can be seen that higher power savings are achieved when the number of clusters is high (compare  $\phi = 10$  clusters of Figure 4b with  $\phi = 200$  clusters of Figure 4e). As soon as we increase the number of clusters, the density of the bars in the figures is higher, meaning that the power savings are increased for most of the TMs.

Finally, Figure 5 reports the gap of the GA compared with LR-EE and 200 clusters. It can be seen that LR-EE outperforms the GA in terms of power savings (negative values) for the majority of the TMs. As a summary, the LR-EE proposed solution is able to obtain better power saving outcomes with respect to the GA for a number of clusters above 10 (see *avg\_gap* column of Table 1), with the corresponding reduction in the computation time.



**Figure 4.** Power savings as a function of the TM Id for GA and LR-EE (TMs are sorted according to their traffic load in ascending order). (a) GA [22], (b) LR-EE  $\phi = 10$ , (c) LR-EE  $\phi = 50$ , (d) LR-EE  $\phi = 100$ , (e) LR-EE  $\phi = 200$ .



**Figure 5.** Power saving GAP of GA vs. LR-EE with  $\phi = 200$ .

**7. Conclusions and Future Work**

In this paper, an ML solution based on LR has been proposed to predict energy-efficient network configurations in SDN, avoiding the execution of optimal or heuristic algorithms at the SDN controller. Thus, a reduction in the computation time derived by the non-execution of these algorithms at the controller is set as objective, along with the improvement in energy savings. Experimental results over a realistic network topology show that, by applying the combination of unsupervised and supervised learning techniques, a notable reduction of power consumption can be achieved by our proposed LR-EE solution compared to the results obtained by energy-efficient ad hoc solutions, with the corresponding significant reduction in the computation time.

Regarding possible future research activities, we work on testing different ML methods to compare them and select the one that provides the best results in terms of power savings

and computation time. We also work on defining a framework that can select the ML technique to use depending on the network topology and other characteristics. In this sense, we are working on using different cluster configuration depending on the network behavior. Finally, we are evaluating the proposed algorithm with larger networks to evaluate the scalability of the solution and how it behaves in emulated environments.

**Author Contributions:** This work was developed by the authors as follows: Conceptualization, J.L.H., J.B. and J.G.-J.; Data curation, J.L.H.; Formal analysis, J.L.H. and J.G.-J.; Funding acquisition, J.B. and J.G.-J.; Investigation, M.J.-L., J.L.H., J.B. and J.G.-J.; Methodology, M.J.-L., J.L.H., J.B. and J.G.-J.; Project administration, J.G.-J.; Software, M.J.-L.; Supervision, J.G.-J.; Validation, M.J.-L. and J.L.H.; Visualization, M.J.-L. and J.G.-J.; Writing—original draft, M.J.-L., J.B. and J.G.-J. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work has been partially funded by the Diputación de Cáceres (AV-7), the project PID2021-124054OB-C31 (MCI/AEI/FEDER, UE), the 4IE+ Project (0499-4IE-PLUS-4-E) funded by the Interreg V-A España-Portugal (POCTEP) 2014–2020 program, by the Department of Economy, Science and Digital Agenda of the Government of Extremadura (GR21133), and by the European Regional Development Fund.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Idzikowski, F.; Chiaraviglio, L.; Cianfrani, A.; López Vizcaíno, J.; Polverini, M.; Ye, Y. A Survey on Energy-Aware Design and Operation of Core Networks. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1453–1499. [[CrossRef](#)]
2. Bianzino, A.P.; Chaudet, C.; Rossi, D.; Rougier, J.L. A Survey of Green Networking Research. *IEEE Commun. Surv. Tutor.* **2012**, *14*, 3–20. Available online: [https://www.researchgate.net/publication/50238130\\_A\\_Survey\\_of\\_Green\\_Networking\\_Research](https://www.researchgate.net/publication/50238130_A_Survey_of_Green_Networking_Research) (accessed on 25 August 2022). [[CrossRef](#)]
3. Strielkowski, W.; Firsova, I.; Lukashenko, I.; Raudeliūnienė, J.; Tvaronavičienė, M. Effective Management of Energy Consumption during the COVID-19 Pandemic: The Role of ICT Solutions. *Energies* **2021**, *14*, 893. [[CrossRef](#)]
4. Rume, T.; Islam, S.D.U. Environmental effects of COVID-19 pandemic and potential strategies of sustainability. *Heliyon* **2020**, *6*, e04965. [[CrossRef](#)] [[PubMed](#)]
5. Cenedese, A.; Tramarin, F.; Vitturi, S. An Energy Efficient Ethernet Strategy Based on Traffic Prediction and Shaping. *IEEE Trans. Commun.* **2017**, *65*, 270–282. [[CrossRef](#)]
6. Nedeveschi, S.; Popa, L.; Iannaccone, G.; Ratnasamy, S.; Wetherall, D. Reducing Network Energy Consumption via Sleeping and Rate-Adaptation. In Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, NSDI'08, San Francisco, CA, USA, 16–18 April 2008; USENIX Association: Berkeley, CA, USA, 2008; pp. 323–336.
7. Galán-Jiménez, J.; Gazo-Cervero, A. ELEE: Energy Levels-Energy Efficiency Tradeoff in Wired Communication Networks. *IEEE Commun. Lett.* **2013**, *17*, 166–168. [[CrossRef](#)]
8. Musumeci, F.; Rottondi, C.; Nag, A.; Macaluso, I.; Zibar, D.; Ruffini, M.; Tornatore, M. An Overview on Application of Machine Learning Techniques in Optical Networks. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 1383–1408. [[CrossRef](#)]
9. Boutaba, R.; Salahuddin, M.A.; Limam, N.; Ayoubi, S.; Shahriar, N.; Solano, F.E.; Rendón, O.M.C. A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities. *J. Internet Serv. Appl.* **2018**, *9*, 16. [[CrossRef](#)]
10. Assefa, B.G.; Özkasap, O. A survey of energy efficiency in SDN: Software-based methods and optimization models. *J. Netw. Comput. Appl.* **2019**, *137*, 127–143. [[CrossRef](#)]
11. Galán-Jiménez, J.; Gazo-Cervero, A. Using bio-inspired algorithms for energy levels assessment in energy efficient wired communication networks. *J. Netw. Comput. Appl.* **2014**, *37*, 171–185. [[CrossRef](#)]
12. Ahmad, S.; Jamil, F.; Ali, A.; Khan, E.; Ibrahim, M.; Whangbo, T. Effectively Handling Network Congestion and Load Balancing in Software-Defined Networking. *CMC-Tech. Sci. Press* **2021**, *70*, 1363–1379. [[CrossRef](#)]
13. Ali, J.; Roh, B.h.; Lee, S. QoS improvement with an optimum controller selection for software-defined networks. *PLoS ONE* **2019**, *14*, e0217631. [[CrossRef](#)]
14. Ali, J.; Roh, B. Quality of service improvement with optimal software-defined networking controller and control plane clustering. *Comput. Mater. Contin* **2021**, *67*, 849–875. [[CrossRef](#)]
15. Xie, J.; Yu, F.R.; Huang, T.; Xie, R.; Liu, J.; Wang, C.; Liu, Y. A Survey of Machine Learning Techniques Applied to Software Defined Networking (SDN): Research Issues and Challenges. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 393–430. [[CrossRef](#)]
16. Chiaraviglio, L.; Mellia, M.; Neri, F. Reducing Power Consumption in Backbone Networks. In Proceedings of the 2009 IEEE International Conference on Communications, Dresden, Germany, 14–18 June 2009; pp. 1–6. [[CrossRef](#)]
17. Lin, Z.; Niu, H.; An, K.; Wang, Y.; Zheng, G.; Chatzinotas, S.; Hu, Y. Refracting RIS-Aided Hybrid Satellite-Terrestrial Relay Networks: Joint Beamforming Design and Optimization. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 3717–3724. [[CrossRef](#)]

18. Lin, Z.; An, K.; Niu, H.; Hu, Y.; Chatzinotas, S.; Zheng, G.; Wang, J. SLNR-based Secure Energy Efficient Beamforming in Multibeam Satellite Systems. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, 1–4. [[CrossRef](#)]
19. Lin, Z.; Lin, M.; Wang, J.B.; de Cola, T.; Wang, J. Joint Beamforming and Power Allocation for Satellite-Terrestrial Integrated Networks With Non-Orthogonal Multiple Access. *IEEE J. Sel. Top. Signal Process.* **2019**, *13*, 657–670. [[CrossRef](#)]
20. Lin, Z.; Lin, M.; de Cola, T.; Wang, J.B.; Zhu, W.P.; Cheng, J. Supporting IoT With Rate-Splitting Multiple Access in Satellite and Aerial-Integrated Networks. *IEEE Internet Things J.* **2021**, *8*, 11123–11134. [[CrossRef](#)]
21. Naeem, F.; Tariq, M.; Poor, H.V. SDN-Enabled Energy-Efficient Routing Optimization Framework for Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2021**, *17*, 5660–5667. [[CrossRef](#)]
22. Galán-Jiménez, J.; Polverini, M.; Cianfrani, A. Reducing the reconfiguration cost of flow tables in energy-efficient Software-Defined Networks. *Comput. Commun.* **2018**, *128*, 95–105. [[CrossRef](#)]
23. Guo, Y.; Wang, W.; Zhang, H.; Guo, W.; Wang, Z.; Tian, Y.; Yin, X.; Wu, J. Traffic Engineering in Hybrid Software Defined Network via Reinforcement Learning. *J. Netw. Comput. Appl.* **2021**, *189*, 103116. [[CrossRef](#)]
24. Guo, Y.; Chen, J.; Huang, K.; Wu, J. A Deep Reinforcement Learning Approach for Deploying SDN Switches in ISP Networks from the Perspective of Traffic Engineering. In Proceedings of the 2022 IEEE 23rd International Conference on High Performance Switching and Routing (HPSR), Taicang, Jiangsu, China, 6–8 June 2022; pp. 195–200. [[CrossRef](#)]
25. Chen, X.; Wang, X.; Yi, B.; He, Q.; Huang, M. Deep Learning-Based Traffic Prediction for Energy Efficiency Optimization in Software-Defined Networking. *IEEE Syst. J.* **2021**, *15*, 5583–5594. [[CrossRef](#)]
26. Chabarek, J.; Sommers, J.; Barford, P.; Estan, C.; Tsiang, D.; Wright, S. Power Awareness in Network Design and Routing. In Proceedings of the IEEE INFOCOM 2008—The 27th Conference on Computer Communications, Phoenix, AZ, USA, 13–18 April 2008; pp. 457–465. [[CrossRef](#)]
27. OpenFlow Switch Specification. Version 1.5.1, Standard, Open Networking Foundation. 2015. Available online: <https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf> (accessed on 25 August 2022).
28. Polverini, M.; Baiocchi, A.; Cianfrani, A.; Iacovazzi, A.; Listanti, M. The Power of SDN to Improve the Estimation of the ISP Traffic Matrix Through the Flow Spread Concept. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 1904–1913. [[CrossRef](#)]
29. Han, J.; Kamber, M.; Tung, A. Spatial Clustering Methods in Data Mining: A Survey. *Data Min. Knowl. Discov.—DATAMINE* **2001**. Available online: [https://www.researchgate.net/publication/238687113\\_Spatial\\_clustering\\_methods\\_in\\_data\\_mining\\_a\\_survey](https://www.researchgate.net/publication/238687113_Spatial_clustering_methods_in_data_mining_a_survey) (accessed on 25 August 2022).
30. Orłowski, S.; Pióro, M.; Tomaszewski, A.; Wessälly, R. SNDlib 1.0—Survivable Network Design Library. In Proceedings of the 3rd International Network Optimization Conference (INOC 2007), Spa, Belgium, 22–25 April 2007. Available online: <http://sndlib.zib.de> (accessed on 25 August 2022).
31. Phillips, C.; Gazo-Cervero, A.; Galán-Jiménez, J.; Chen, X. Pro-active energy management for Wide Area Networks. In Proceedings of the IET International Conference on Communication Technology and Application (ICCTA 2011), Beijing, China, 14–16 October 2011; pp. 317–322.