This article has been accepted for publication in IEEE Transactions on Green Communications and Networking. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TGCN.2023.3282326

Digital Twin Enriched Green Topology Discovery for Next Generation Core Networks

Kübra Duran, *Student Member, IEEE,* Berk Canberk, *Senior Member, IEEE.*

***Abstract*—Topology discovery is the key function of core network management since it utilizes the perception of data and mapping network devices. Nevertheless, it holds operational and resource efficiency complexities. For example, traditional discovery cannot perform predictive analysis to learn the network behavior. Moreover, traditional discovery periodically visits IP ports without considering the utilization levels, which leads to high resource usage and energy consumption. Hence, it is necessary to integrate intelligent methods into traditional discovery to deeply understand the behavioral pattern of a core network and recommend action to avoid these intrinsic complexities. Therefore, we propose a Digital Twin (DT) enriched Green Discovery Policy (DT-GDP) to serve a green discovery by using the increased intelligence and seamless assistance of DT. DT-GDP jointly uses the outputs of two modules to calculate the total energy consumption in Watts. In the energy module, we consider the service power, idle state power, and the cooling power of an IP port and derive a novel energy formula. In the visit decision module, we use Multilayer Perceptron (MLP) to classify the IP ports and recommend visit action. According to experimental results, we achieve a significant reduction in the visited ports by 53% and energy consumption by 66%.**

***Index Terms*—energy efficiency, sustainable network discovery, multilayer perceptron, digital twin networks**

I. Introduction

In the last few years, energy efficiency in core network discovery has become the key consideration for Internet Service Providers (ISPs) due to the rising energy costs and the critical environmental effects of network devices. At this point, intelligent and resource-aware discovery approaches become crucial to serving green and reliable network management by considering the network convergence time. However, traditional discovery does not consider energy consumption issues and requires a high amount of resources, for instance, high runtime, CPU, and memory usage. Also, due to the lack of predictive and prescriptive methods, traditional discovery cannot provide fast convergence time. Although today's ISPs start with searching for new ML/DL (Machine Learning/Deep Learning) integrated approaches with the focus on green and highly reliable network services [1], no effective solution has been found yet. Furthermore, the Sustainable Development Goal (SDG) 7.3 of the United Nations' urges energy efficiency to double the global energy efficiency rate by 2030 [2] along with many vertical domains, such as industrial applications, ICT, etc. For this, we note that increased intelligence integrated with virtualization technologies, such as Augmented Reality (AR)/Virtual Reality (VR), Digital Twin Networks (DTNs), Hologram Communication, etc., have advantages in the reduction of physical resource usage and thus contribute for green network services [3], [4].

K. Duran is with the Department of Computer Engineering, Istanbul Technical University, Istanbul 34467, Turkey (e-mail: durank18@itu.edu.tr)
B.Canberk is a professor within the School of Computing, Engineering and Built Environment, Edinburgh Napier University, Edinburgh EH10 5DT, UK (e-mail: B.Canberk@napier.ac.uk)

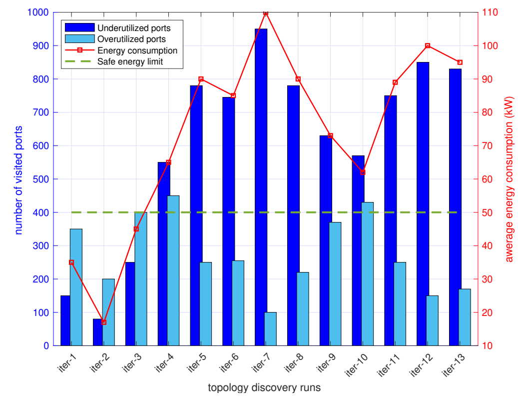


Fig. 1: Discovery service resource usage and energy consumption characteristics

A. Main Challenges in Core Network Discovery

1) *Lack of smart monitoring and controlling mechanism to deploy prescriptive models and run what-if-analysis:* In today's ISPs, traditional discovery is driven manually. However, this method is insufficient to take an insight into the future state of the network behavior to recommend actions and run a what-if-analysis to avoid sudden discovery of performance degradation.
2) *Energy efficiency:* Port-level energy consumption of network discovery impacts the total energy consumption of a router [5]. Moreover, the long-running period and putting extra management traffic on the links increases the energy consumption of discovery. This is because the high load on the links for a long time affects the CPU usage, response time, and thus the energy consumption. For example, in Fig. 1, traditional complete discovery working on a medium-sized ($\sim$ 40 routers) IP core with the number of visited ports and average energy consumption can be seen. Although the average energy consumption limit for a medium-sized topology is around 100kW, traditional complete discovery exceeds this limit due to the periodic visit to all underutilized and overutilized IP ports. Besides, overutilized ports could lead to quality of service-related degradations in network devices. This is because overutilization puts a

high workload on the network links, which leads to the heating of processing devices at the same time.

These mentioned problems require intelligent management abilities to deliver a more green and resource-aware network discovery. Digital Twin (DT), as an emerging digital technology, brings the virtual twin of any physical entity of real-world equipment. ZSM*-ETSI* is examining the application of ML/DL to realize automated network management by referring to DT as an increased intelligence procedure [6]. Besides, ITU-T* also proposes a high-level functional architecture [7] for ML integrated next-generation networks by considering DT mirroring. It is seen that DT integration models are strong enablers for future telecom networks that tend to have increased complexity and comprehensive learning abilities alongside proactive management needs.

B. Why Do We Need DT Technology for a Green Core Network Management?

The main reasons why we propose DT integration to network discovery service are as follows: **(i)** DT serves real-time remote monitoring and control along the entire lifecycle of discovery service, which enables deploying predictive and prescriptive analytics and running what-if-analysis to learn more about the network behavior (addressing Challenge 1), **(ii)** In contrast to traditional discovery services, DT can support services with a direct linkage to network operational data rather than lastly updated IP inventory. Lastly updated IP inventory becomes outdated as time passes until the next discovery service cycle starts. For this reason, DT-integrated discovery results in more accurate topology information via smart observation and control (addressing Challenge 1). **(iii)** Thanks to its virtualization feature and seamless assistance, DT provides resource and energy saving for discovery service. This is because, in a DT-integrated discovery, there is no need to access all physical IP ports, which reduces the total CPU and memory usage. For this, less amount of resource usage will result in reduced energy consumption for the discovery service while maintaining highly reliable topology information that surpasses the traditional partial discovery methods (addressing Challenge 2), **(iv)** The configuration settings of a network device include the information on how to obtain the operational data regarding the device. Unlike the traditional methods, DT integrated discovery service does not require additional configurations in the perception of data from different types of network devices. It eliminates the extra data gathering step with a contribution to lowering energy consumption (addressing Challenge 2), **(v)** In contrast to traditional discovery services, DT supports the implementation of various Machine Learning methods to predict the unnecessary visits and thus decrease the energy consumption of discovery (addressing Challenge 2).

As a result of these advantages, we propose to integrate Digital Twin networks into the network discovery to manage the discovery process in a more efficient way regarding the network resource usage and environmental effects. Although the DT seems like a separation of layers as in Software Defined Networks (SDN) technology, there are significant differences in terms of network discovery. For instance, the SDN control plane implements a centralized controller to discover the configuration information of network devices, while DT aggregates and analyzes the configuration information from various sources. In addition, DT makes it possible to implement AI/ML algorithms and identify patterns and anomalies. However, the SDN control plane cannot achieve such smart methods to deeply understand the network dynamics and take smart actions in network discovery. In summary, the DT control tower uses a virtual representation of the IP core and learning capabilities to enable an energy-efficient network discovery in contrast to the centralized SDN control plane.

*Abbreviations:** ZSM: Zero-touch Network and Service Management, ETSI: European Telecommunications Standart Institute, ITU-T: International Telecommunications Union Telecommunication Standardization Sector

C. State of the Art

In the current literature, there are significant attempts to integrate DT technology into network management and control systems to enhance communications services. For example, in [8], enabling the IoT applications over the sixth generation (6G) networks is discussed to manage and optimize 6G edge networks. Moreover, [9] discusses 6G Digital Twin Networking (DTN) from both data and communication perspectives by resulting in a reference architecture for interfaces between the layers to form a closed-control loop. [10] gives a broad taxonomy of DT and 6G integration. In addition, the authors examine the potential challenges from a data management perspective for 6G-oriented services. Furthermore, [11] examines the implementation of network slicing by using DT regarding the Industry 4.0 and 5G applications. According to the simulation results, the authors ensure that the DT model accurately mirrors the network behavior and predicts end-to-end latency under various environmental conditions. Regarding the DT modeling and applications, [12] serves as a comprehensive survey highlighting the DT and product lifecycle relations. This study mentions the Digital Twin and Digital Shadow terms for the current DT modeling attempts. In addition, the DT design phase is referred to as the production system design to optimize iteratively, analyze and evaluate virtually.

On the other hand, the current literature is categorized into two parts from the network discovery perspective regarding the node visit behavior; TPD (Traditional Partial Discovery) and TCD (Traditional Complete Discovery). In TPD, a subset of topology is chosen and visited instead of periodical visits. However, its usage has not yet been widespread in the telco industry. In our previous work [13], we propose a device-level TPD approach by using Hidden Markov Model to estimate the possible devices which need to be discovered in an ISP network. Furthermore, in [14], a new partial discovery scheme is proposed to increase the accuracy rate of neighbor discovery for a transmit-receive pair in cellular networks. In [15], a new topology inference method based on a subset structure is proposed to achieve decreased packet delay to construct a holistic network view. In [16], a new partial discovery approach is introduced for multi-UAVs in ad-hoc networks to solve network reconstruction problems by reducing the coverage time and increasing the reconstruction accuracy. Furthermore, in [17], an adaptive topology reconfiguration method is proposed for DNs (Distribution Networks) to handle reliability challenges in DN lines.

In [18], a complete discovery is proposed for an SDN deployed scenario to eliminate the redundant messages for the central controller. Likewise, in [19] and [20], new complete discovery mechanisms are proposed depending on the OpenFlow protocol to improve the discovery time. In [21], a complete neighbor discovery scheme is examined to solve the mobility and the battery limitation dilemma in IoT networks. Besides in [22], energy efficiency is considered for underwater acoustic sensor networks (UANs) discovery due to the limited lifetime of the underwater vehicles.

As seen in Table-I, existing state-of-the-art TPDs and TCDs are able to serve accurate topology information for different networks. However, only IoT cases are energy-oriented. Moreover, IP core-based studies do not consider the underutilized and overutilized ports regarding energy consumption. At this point, the high energy consumption of discovery stemming from underutilized ports remains serious. Although several examples of DT developments for network management are listed above, none of them focuses on network discovery. To the best of our knowledge, this is the first study to develop a sustainable topology discovery for an IP core network by using DT integration. Thereby, our research question for this study is *"How can we serve an intelligent, and energy-aware discovery for IP core considering the specified energy definition, which uses the parameters; port usage, bandwidth requests, and capacity of the ports ?"* To address this; we propose the use of DT technology by taking advantage of its resource efficiency and prescriptive analytics feature.

D. Contributions

We summarize the contributions of this study as follows:

- We construct the digital twin layer of the physical IP core to maintain more efficient discovery management in which descriptive, predictive and prescriptive analytics are enabled to learn the network behavior and perform what-if-analysis.
- We model the total energy consumption of discovery as a function of the average number of visited IP ports and their individual energy consumption. With the integration of DT, our energy consumption model embodies the virtual neighboring circumstance of the physical IP ports.
- As an agile brain of our solution, we use Multilayer Perceptron integrated with the DT capabilities to perform an action recommendation for a port-level discovery.

The rest of the paper is organized as follows. In Section II, DT-GDP architecture is given in detail. Section III is devoted to the performance evaluation of DT-GDP. Finally, Section IV concludes the paper.

II. DT-GDP Architecture

A. Physical IP Core

We consider a physical IP core topology that serves broadband services to regional internet users. At the top of the broadband access topology, there is an IP/MPLS (Internet Protocol/Multi Protocol Label Switching) core, which makes it possible to carry different types of traffic and access available services. In this core, there are P (provider) and PE (provider edge) routers. In our physical twin topology, we use single-rack Cisco CRS-1 core router specifications to calculate energy consumption. For example, we use consumed power as approximately 106 W at full load and 92 W at idle state [5]. Fig.2 describes the general application architecture of TPD and TCD on a sample IP core. As the figure shows, the discovery agent takes information from the IP core via IP access methods, specifically snmpwalk (a Simple Network Management Protocol application) and ICMP (Internet Message Control Protocol), and then constructs the physical and logical inventory regarding the network topology. TCP and TPD discovery approaches are implemented within the discovery agent by jointly using the snmpwalk and ICMP outcomes. Therefore, we have adopted a partial discovery approach in DT-GDP by implementing snmpwalk and ICMP. We have gathered the port information of the network devices' via snmpwalk. After retrieving the network devices' diagnosis information, we used ICMP to test the connectivity status of the network devices. We have used ICMP while encountering missing diagnosis information within the snmpwalk outputs.

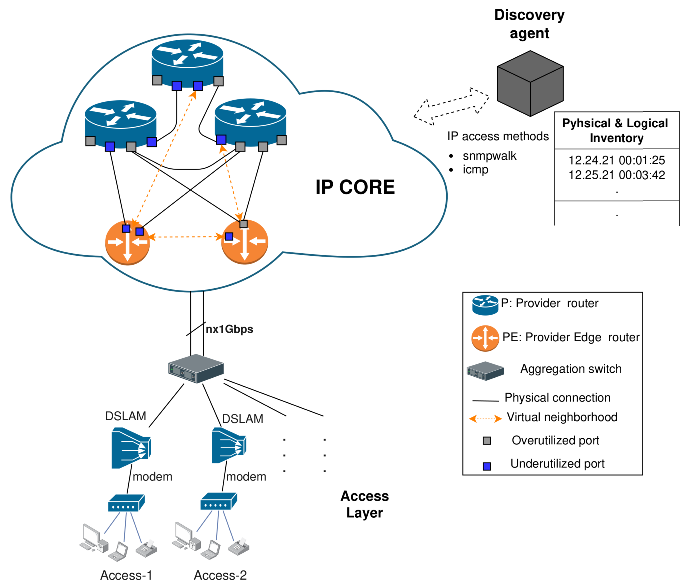


Fig. 2: Traditional discovery service topology working on a sample IP core layer

B. Twin IP Core

As shown in Fig. 3, DT-GDP starts with processing the real-time, historical, and physical asset dataset of the physical topology. After it constructs the virtual twin layer in which the DT model is formed using the processed asset set. After that, the intelligence feature of the twin IP core layer is activated within the *Visit Decision Module*, and the specific energy consumption formula is formed within the *Energy Module*. The outputs of these two main modules predicted visit behavior and per port energy consumption, are combined

This article has been accepted for publication in IEEE Transactions on Green Communications and Networking. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TGCN.2023.3282326

TABLE I: Comparison of Proposed Green Discovery with Existing Works

Literature	Discovery approach	Infrastructure	Number of nodes visited	Energy consumption	Topology construction accuracy	Average node availability
[13], [14], [15]	partial	IP core, Cellular	low	high	high	low
[16], [17]	partial	UAV, DN	low	high	high	high
[18], [19], [20]	complete	SDN	high	high	high	low
[21], [22]	complete	IoT, UAN	high	low	high	low
Our work	partial	IP core	low	low	high	high

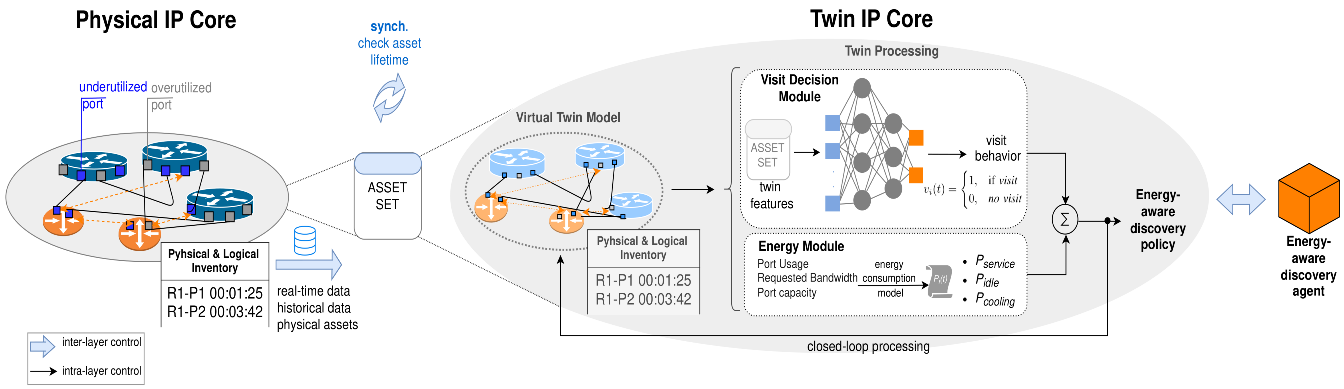


Fig. 3: Proposed structure for Digital Twin Enriched Green Discovery Policy, DT-GDP

to generate an energy-aware and intelligent discovery policy. The block diagram for the twin processing step is given in Fig. 4. Furthermore, the energy-aware discovery policy includes a closed-loop feedback system that feeds itself with the learned parameters. This mechanism sustains the topology information accuracy to the desired levels. In addition, synchronization is performed to check for the asset dataset lifetime between the physical IP core layer and the twin IP core layer. At this point, propagation and processing delays are two issues in synchronization related to the communication concerns in DT networking. As our main focus is developing a discovery service-wide energy consumption model by considering the service and cooling power, the communication issues remain out of scope for this study. Therefore, we perform a manual synchronization frequency set approach as given in [23]. The details of DT-GDP are explained in the below sections, and the notations used within this study are given in Table II.

1) Energy Consumption Model: Generally, a router's power consumption is related to its processing power, the total number of ports, the type of these ports, and the activity on those ports. For instance, a router with more or higher speed ports consumes more power due to increased data flow in both ingress and egress directions. In this circumstance, the power consumption of a router port is lower than the total power consumption of this router. On the other hand, the port speed and the type of port connection (fiber optic, Ethernet, etc.) change, and the power consumption of a port can reach high levels. Considering this, we work on Cisco CRS-1 routers with identical port types to realize and test our theoretical work in a clarified manner.

Before starting the port-based energy model development, we need to find a way to relate the port features and find interdependencies between them. For this, we work on the MPLS dataset used in [13] by extracting the Cisco CRS-1

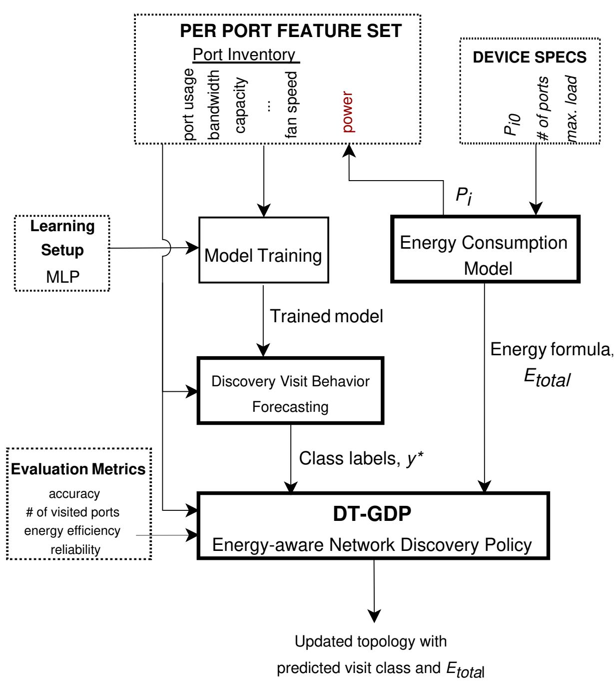


Fig. 4: Block diagram for twin processing step, realized in Twin IP core Layer

router rows and their features (mainly including the energy consumption opposed to the given study) as the columns. After, we perform correlation analysis for each feature to see how the features affect each other and energy consumption on this dataset. We use $corrcoef()$ method in MATLAB R2020b© to get the correlation coefficients for the features. We feed this method with the feature matrix consisting of

TABLE II: Nomenclature

Notation	*Explanation*
F	Asset set for twins
n	Total number of routers
m	Total number of ports
k	Total number of ports in a router
E_{total}	Total energy consumption in network discovery
$P_{service}$	Energy consumed when the discovery service in run
P_{ij}	Energy consumed by the j^{th} port of an i^{th} router
P_{imax}	Energy consumed by router i at full load
P_{idle}	Energy consumed by port a router at idle state
P_c	Cooling power for a router
P_{ci}	Cooling power for a port
v_i	Visit coefficient
γ	Port usage
β	Requested bandwidth
C	Capacity of the port
a	IP port access rate
$\hat{c_i}$	Output label by layer i
$\hat{C}^t$	Output vector at time t
c_e	Cross-entropy value
R	Total reliability value
r_{ij}	Reliability constant for action j in case i
c_{ij}	Cost constant for action j in case i

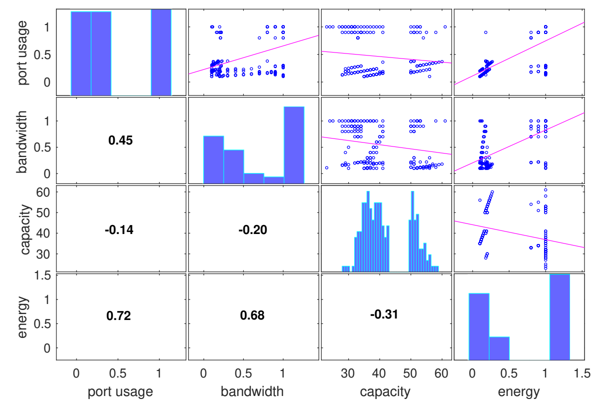


Fig. 5: Energy consumption model parameters correlation matrix

the Cisco routers in rows and the respective feature values in columns. The resultant correlation matrix is given in Fig. 5. This figure shows that three-port features have a nonzero correlation with the theoretical power. Moreover, the lower triangular side of the correlation plot shows the Pearson's Correlation Coefficient (PCC) values for linear correlations. By looking at these correlation values, it seems that the port capacity has a negative correlation value that is -0.31, with the consumed energy. This correlation result is a possible case with the constant service requests. Moreover, regarding the correlation values, the requested bandwidth and port usage seem to have positive correlations, with $0.68, 0.72$ values, respectively, with the consumed energy. Furthermore, this case is also sensible regarding the increased service demands. As the nonzero correlation means the features affect each other, they can be considered all in one to define a new metric: energy consumption. For this reason, we form a specific energy consumption formula for our energy-aware discovery by using port usage (γ), requested bandwidth (β), and capacity of the port (C).

In our energy consumption model, we take into consideration the energy consumed by the discovery service, the energy consumed at the idle state of a router, and the cooling power of the router. The energy required for port visits constitutes the discovery service power. The idle state power arises when there is no active running service on a router. In addition, the cooling power of a network device, a router in our study, is related to the given discovery service power. By considering these terms, we form the energy consumption of the j^{th} port in the i^{th} router as:

$$P_{ij} = P_{ijService} + P_{ij0} + P_{ijc}, \forall_{i,j} \tag{1}$$

where $P_{ijService}$ is defined as the energy consumed when the discovery service is running at full load. P_{ij0} is the idle state energy consumption, and P_{ijc} is the cooling power of a port. The energy consumption of discovery service at full load is calculated as:

$$P_{ijService} = \frac{\beta}{C} . \frac{P_{imax}}{k} . \gamma \tag{2}$$

where β and C values are in Gbps, and γ is in percentage. We note that β stands for requested bandwidth for the management interfaces of a router. For this reason, $\frac{\beta}{C}$ can be interpreted as service request rate.

Moreover, γ is the port usage rate in percentage that is the actual management port utilization rate. The maximum value for this utilization rate is equal to 1. Furthermore, the requested bandwidth value plays a significant role in the resource usage of the management interfaces. In other words, if there arises a high amount of bandwidth request, CPU usage, processing capability, response time, current device temperature, and thus the consumed energy values are affected. For this reason, we not only multiply the port power by (γ) but also by ($\frac{\beta}{C}$) to hit the dense management traffic arrival cases. Also, P_{imax} in the nominator stands for the max power consumed by the router i at maximum load. We take this value from the Cisco CRS-1 router datasheet equal 106Watt. We assume fair port usage for active services in the maximum load case. On the other hand, when the discovery service and all other services are not running, the router will be in an idle state. Hence, there arises minimum power consumption, called idle state power consumption. It is calculated as:

$$P_{i0} = \frac{P_{iidle}}{k} \tag{3}$$

where k stands for the number of ports regarding the related router. In addition, from the energy consumption perspective, there arises an additional consumption term for a router, that is, the cooling power. As the routers generate heat when they are operating according to the load, the energy required to cool that arisen heat is in non-negligible levels. In addition, according to [5], the required cooling power is equal to the given service power. Therefore, we take into consider cooling power of a router when modeling the discovery service to converge to a more realistic energy consumption model. In this

This article has been accepted for publication in IEEE Transactions on Green Communications and Networking. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TGCN.2023.3282326

case, the total energy consumption of a router for discovery service will be doubled due to the cooling power by reaching higher levels. That's why we take into consider the cooling power in our theoretical modeling. Hence, we define the per port cooling power as:

$$P_{ijc} = (P_{ijService} + P_{ij0}).a \tag{4}$$

where, a is the average access rate to the IP ports and $\in [0, 1]$. Access rate means, if the requested service is high, the access rate will be high for these ports. The calculation of a is given below:

$$a = \frac{1}{k}\sum_{i=1}^{k} v_i \frac{\beta}{C} \tag{5}$$

where v_i is the visit coefficient that is decided by *Visit Decision Module*. In the first run of the DT-GDP, v_i values for all ports are initialized as one, which is the maximum value. In the upcoming iterations, it is decided in the *Visit Decision Module* and feedbacked to the virtual twin model.

2) Visit Behavior Decision: The visiting decision for IP ports is made by using the twin layer features. As given in Fig. 6, the proposed DT-GDP uses Multilayer Perceptrons (MLPs) in Visit Decision Module to find out the discovery visit behavior. Since some of the IP port features in the dataset are strongly correlated with the energy consumption value and the visit behavior, it might not be enough to use statistical models or optimization approaches. Also, by looking at the performance results of the simulated prediction methods, we choose MLP as the brain function of DT-GDP as achieving the highest accuracy. Thus, MLP takes $X^t \in R^{dx25}$ as the input, where d stands for the number of instances, and 25 stands for the number of features in the twin layer. In our case, as the number of instances will be equal to the total number of ports in the inventory for the time t, we say that $d = m$.

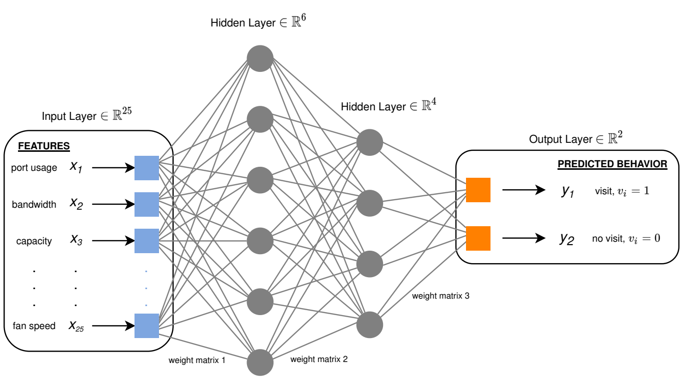


Fig. 6: Internal structure of proposed Visit Decision Module

Furthermore, by looking from the derived energy model perspective, we see that there arise two classes with *visit* and *no visit* labels to be mapped with the actual overutilization and underutilization status of the ports. Here, predicted class values are achieved by using all features listed in the dataset. As the features have interdependency with each other, using the result of this multi-featured effect to plan the port visits give more significant results rather than using only the utilization rate feature. With this approach, DT-GDP maintains a revealing topology more applicable for what-if-analysis. The performance results of one-feature and multi-feature cases are given in Section-IV.D in detail. As a result, we train our model on MLP in order to solve a two-class classification problem. We use randomized 5-fold cross-validation to decide the MLP architecture and select (6,4) 2-layered architecture, which is given in Fig. 6. In this MLP architecture, we use $sigmoid$, $\sigma(.)$ activation function in the label decision step, which outputs the labels by:

$$\hat{c_h} = \frac{1}{1 + exp(\sum_{j=1}^{d} w_{hj} x_j + w_{h0})} \tag{6}$$

where, $h \in \{1, 2, 3\}$ stands for the 2 hidden layer indexes and the output layer index respectively, and $\hat{c_h} \in \{0, 1\}$ stands for the class labels in our 2-layered MLP architecture. Besides, $w_{h(.)}$ is the weight vector for the layer h^{th} layer, and it is updated according to the loss function output. As the loss function, we use binary cross-entropy (c_e) which is given below:

$$c_e = c_h \log \hat{c_h} + (1 - c_h) \log(1 - \hat{c_h}) \tag{7}$$

where, c_h values are the actual class labels, and $\hat{c_h}$ values are the predicted class labels. In the implementation of MLP, the cross-entropy is calculated for each of the predicted class labels and the update rule for stochastic gradient descent algorithm is realized by trying to achieve minimum cross-entropy value in total. According to the output of MLP that is, $\hat{C^t} \in R^{dx1}$, we interpret the visit behavior of the j^{th} port in the i^{th} router as given below:

$$v_{ij}(t+1) = \begin{cases} visit, & \text{if } \sigma(w^T x) = 1 \\ no\ visit, & \text{if } \sigma(w^T x) = 0 \end{cases}$$

By using MLP outputs, DT topology is updated by considering only the ports with the $visit$ predicted label. More specifically, if a port visit behavior is predicted as $visit$, the asset set of this port is updated. If it is $novisit$, then there is no update for this port. Moreover, the visit decision is performed for all ports at each time step. This is because the visit decision is a kind of service for the twin layer that makes DT up-to-date. To calculate the total energy consumption during the discovery service, we use:

$$E_{total} = \sum_{i=1}^{n} \sum_{j=1}^{k} v_{ij} P_{ij} \tag{8}$$

where n stands for the number of routers, and k stands for the number of ports in a router. Thereby, the outputs of both the Visit Decision Module and the Energy Module, respectively, are used to traverse all ports in all routers contributing to the total energy consumption. This approach basically depends upon summing together the visited ports' energy consumption by excluding the energy consumption of the non-visited ports.

As seen in Alg.1, DT-GDP comprises of two functions, such as *visitDecision* and *energyCalculation*. Each of these functions covers the newly designed modules within the twin IP core layer. Before calling these functions in the main body of DT-GDP, the initialization step is performed by setting the visit behavior for all ports to 1, setting the weight values between the layers of MLP in the interval of (-0.01,0.01) in a randomized manner, and synchronizing the asset set of the physical-digital twin. The *energyCalculation* function is called in line-15 in Alg.1. It realizes our energy consumption model that is derived in the Energy Module and calculates the total energy consumption of a port located on a core IP router in line-10. After that, the visit behavior forecasting is performed by using MLP between line-12 and line-21 in Alg.1. Moreover, between line-17 and line-19, cross-entropy values are calculated to use in the updating weight values. The updating rule is performed by adding the Δw values to the current weight values. In line-18, Δw values are calculated by using learning rate (α), cross-entropy values, and the input set. After getting the MLP outputs, we interpret them to form the visit behavior decision as given in line-22. Eventually, the outputs of the two functions, visit behavior and the consumed energy by each port, are used to calculate the total energy consumption of a discovery service in line-23.

Algorithm 1 DT-GDP

Require: F, $X^t \in R^{dx25}$, n, m, k, P_{idle}, P_{max}, *MLP parameters*
Ensure: $\hat{C}^t \in R^{dx1}$, ${v_{all}}^{t+1}$, E_{total}

1: **initialization**
2: ${v_{all}}^{t+1} \leftarrow 1$
3: $w_{all} \leftarrow rand(-0.01, 0.01)$
4: Synch. asset set, F of physical twin
5: **function** energyCalculation(β, C, γ, ${v_{all}}^{t+1}$, P_{idle}, P_{max})
6: **foreach** router i **from 1 to n**
7: **foreach** port j **from 1 to k**
8: Calculate average port access rate, a_{ij}
9: Calculate $P_{ijService}$, P_{i0}, P_{ijC}
10: $P_{ij} \leftarrow P_{ijService} + P_{i0} + P_{ijC}$
11: **end**
12: **function** visitDecision(X^t, *MLP parameters*)
13: **while** in time step t **repeat**
14: **foreach** $x_i^t \in X^t$ in random order
15: **for** h **from 1 to 3**
16: Predict class labels, $\hat{c_h}$
17: $c_e \leftarrow \prod_h (\hat{c_h})^{c_h}(1-\hat{c_h})^{1-c_h}$
18: $\Delta w_{all} = \alpha c_e x_i^t$
19: $w_{all} \leftarrow w_{all} + \Delta w_{all}$
20: **until convergence**
21: **end**
22: Interpret $\hat{C}^t$ and update ${v_{all}}^{t+1}$
23: Calculate E_{total}
24: **end** =0

III. PERFORMANCE EVALUATION

A. Dataset Description

We used an MPLS service dataset that is provided by a telecom service provider in Turkey. We have worked on this dataset before [13], but did not consider the energy consumption values. As we work on Cisco CRS-1 routers, we crop the dataset to extract only the CRS-1 specific information. As a result, our final dataset consists of 500 samples, each of which stands for an IP port of a core router Cisco CRS-1. In the training process, we have implemented 5-fold cross-validation to have a more robust trained model by tuning the hyperparameters. In the implementation of sparse topology, we take the samples from the same dataset by using the rate of 35%. In addition, each sample in the dataset has a 25 x 1 feature vector with the features of {port usage, requested bandwidth, the capacity of the port, total CPU load, memory usage, fan speed... etc. }. The full list of these features and labels is given in Table III. Also, the distribution of port usage values over two classes is given in Fig. 7 with the average values shown as the dashed lines. We partition the data into three parts, such as 70% as training data, 15% as test data, and 15% as validation data to use in the learning and prediction steps.

TABLE III: Features and Labels Summary

Features	
Port features (16 features)	Router features (9 features)
-port usage (min/max/avg)(%)	-avg number of active links
-requested bandwidth (Gbps)	-number of ports
-capacity	-cpu load (%)
-status (up/down)	-memory usage (%)
-fan speed (%)	-load (min/max/avg)
-temperature (C)	-location
-theoretical power (Watt)	-region
-buffer state (%)	-status
-packet drop count	
-mean packet size (bytes)	
-ingress packet rate	
-egress packet rate	
-connection percentage	
Labels	
-underutilized	
-overutilized	

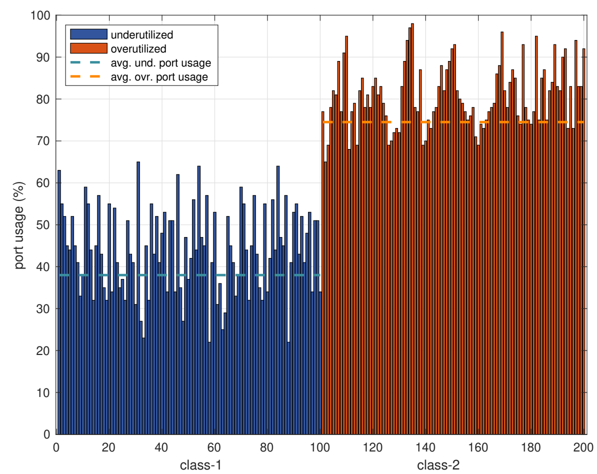


Fig. 7: Port usage distribution of two classes; Class-1 Underutilized ports, Class-2 Overutilized ports

B. Experimental Setup

As seen in Fig. 8, we carried out model training on MATLAB R2020b© environment using Deep Learning Toolbox. We have set up the simulation modules in Deep Network Designer according to our two-layered MLP structure working on an Ubuntu 16.04 desktop. We use MATLAB Production Server to test and store the models and handle DT version management. The production servers are integrated with the Microsoft's Azure Digital Twin Platform. According to the most accurate models, we update the current DT, which is available to network services. The data communication between the physical IP core and DT IP core is maintained by using REST calls. Also, database accesses are performed by using SQL queries. All the simulation parameters are given in Table IV.

TABLE IV: DT-GDP Core Network Discovery Parameters

Parameters	Values
Topology Parameters	**Total Number of routers in Sparse and Dense Topologies:** {85, 200} **Total Number of ports in Sparse and Dense Topologies:** {175, 500} Including: Overutilized and underutilized ports
Router Parameters	**Implemented router:** CISCO CRS-1 **Capacity:** 4-slot, single-rack **Full load energy consumption,** P_{max}**:** 106W **Idle state energy consumption,** P_{idle}**:** 92W **Port density:** 5x100G
Learning Model Parameters	**MLP (6,4) 2-layered architecture** **Optimizer:** sgdm **Cross-validation:** 5-fold **Loss function:** Binary cross-entropy **Learning rate update period:** 3 **Maxepochs:** 40 **Mini batch-size:** 125

C. Targeted Parameters & Baselines

In this simulation, we aim to investigate the results of DT-GDP, for the parameters: (i) number of ports visited, (ii) total energy consumption of discovery service, (iii) average resource consumption, and (vi) total cost value. We compare the performance of DT-GDP with TPD and TCD discovery approaches. We test a set of prediction methods as given below and compare the classification results with MLP.

1) Naive-Bayes: In the Naive-Bayes model, we use the Gaussian approach assuming that the likelihood of features follows a normal distribution.

2) Quadratic Discriminant: In the Quadratic Discriminant model, we try to find a non-linear boundary between the two classes by using a quadratic kernel.

3) KNN: In the KNN model, we use fine KNN with the euclidean distance metric and equal distance weights when calculating the decision criteria.

4) Decision Tree: In the running of the Decision Tree model, we form decision surfaces by using Gini's s Index as the splitting criteria.

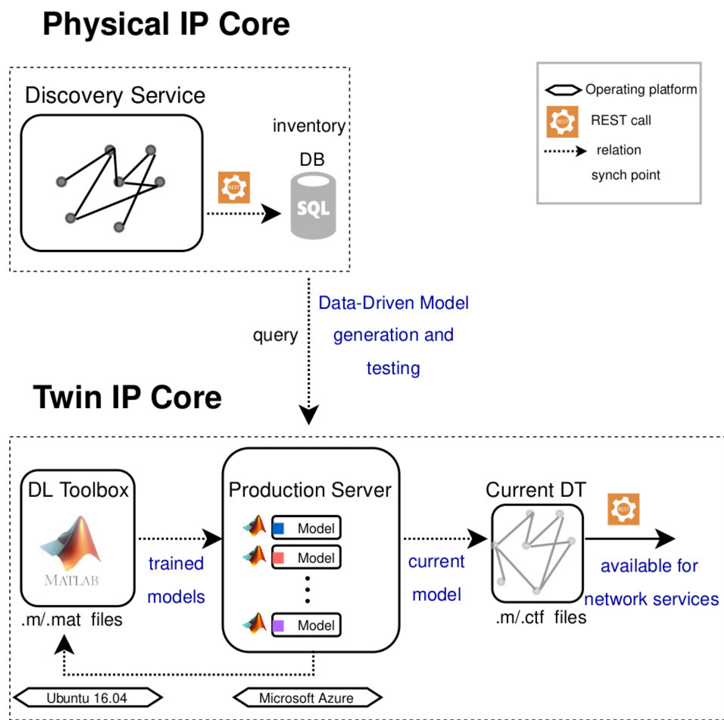


Fig. 8: System architecture of the DT realization process

5) SVM: In the implementation of the SVM model, we use kernel type as *poly*. In addition, other hyperparameters for this model, such as regularization term and the kernel coefficient are decided according to the cross-validation results.

D. Performance Comparison

1) Visited Number of Ports-Energy Consumption Relation: Firstly, we perform DT-GDP on sparse and dense topologies to measure the visited number of ports and the total energy consumption in Watts. We run the simulation in parallel for a non-stop discovery and observe the results for 20 minutes of simulation time. At this point, we choose such a topology sample to run a discovery service that will take 20 minutes on average. Namely, the simulation of discovery starts at the $t = 1$ and ends at the end of the twentieth minute. We show these steps in Fig. 9 to make a clear observation of a visited number of ports and energy consumption. We compare the results with TCD and TPD methods. As seen from Fig. 9 DT-GDP in sparse topology, which is shown as a blue line, visits 40% fewer IP ports than the TCD and TPD for a sparse topology. Furthermore, for a dense topology, the visiting behavior of DT-GDP occurs as $\sim 55\%$ less compared to TCD and TPD. For example, if we look at the green and orange colored rectangles in Fig. 9, we see that the average visited a number of ports with DT-GDP is much closer to the sparse topology results. This means that DT-GDP surpasses the other methods, even if in the dense topology scenarios. Moreover, if we calculate the average energy consumption regarding the colored rectangles, we see that DT-GDP consumes 66% less energy in a dense topology. The main reason for the

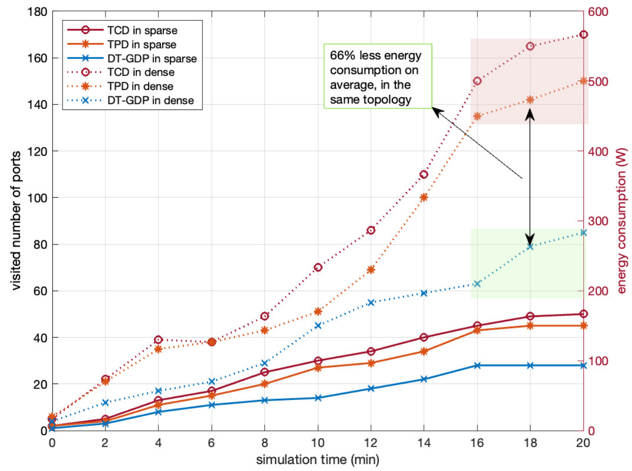


Fig. 9: Visited number of ports and energy consumption value during simulation time

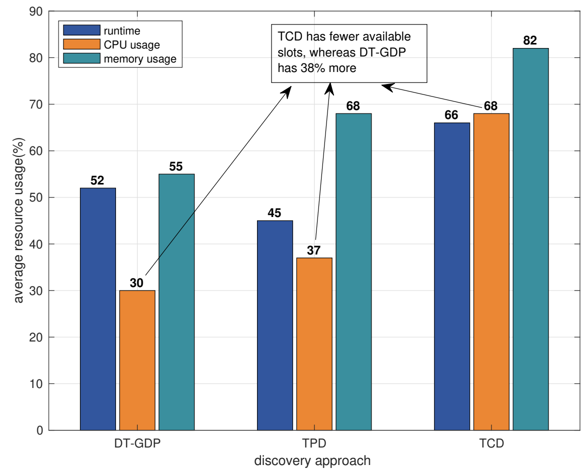


Fig. 10: Average resource usage comparison

TABLE VI: DT-GDP Relative Cost Analysis

Sparse Topology	**Dense Topology**
27% ↓	45% ↓

success of DT-GDP comes from making a distinction between overutilized and underutilized ports (in Alg.1 line-22) and eliminating unnecessary visits to IP ports.

2) Average Resource Usage: We examine the average resource usage of the discovery methods by measuring the total runtime, CPU usage, and memory usage of TPD, TCD, and DT-GDP. As seen from Fig. 10, although DT-GDP is not the fastest discovery method, it occupies the least CPU and memory resources in percentage when compared to TCD and TPD. The main reason for this is that even though it runs ML algorithms, the snmpwalk to a high number of network devices puts more load on the network devices compared to the data-driven algorithm run. Correspondingly, the implementation of DT (assuming that DT communication issues are handled) has resulted in a more effective method to eliminate the unnecessary snmpwalk process to many devices. As a result, it has a wider range of available resource slots, as shown in the figure with green two-sided arrows. The main reason for the resource slot availability is that we deploy the data-driven digital twin on the cloud by using the capability of Matlab Production Server to be deployed on cloud platforms. With this, physical production servers' management cost and resource usage are eliminated from the overhead of DT deployment.

TABLE V: DT-GDP loss matrix in terms of discovery service relative cost evaluation

		Actions	
		No visit	*Visit*
Cases	$y_{ij}=0$	c: 0	c: P_{ij}
	$y_{ij}=1$	c: $-P_{ij}$	c: 0

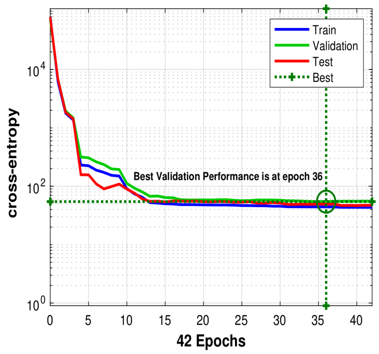
3) Relative Cost Analysis-Energy Consumption Relation: The relative cost value is the measure of how much the proposed method diverges from the ideal case in terms of energy consumption. The higher the relative cost-value means, the higher the extra energy consumption that is not likely to occur in the ideal case where all visits are assumed to be performed correctly. We construct the loss matrix of DT-GDP as given in Table V. In this matrix, we put the actual class labels, that are y_{ij}' s, to imply the row elements. On the contrary, we put the actions to be taken according to the predicted class labels as the column indicator. In the loss matrix, c values stands for cost rate for the case-action pairs. We define the values for c according to our discovery decision problem.

For the cost rate, c, we put positive P_{ij} values if the real class label of a port is zero and it is misclassified because of being an unnecessary visit in reality. Likewise, we put negative P_{ij} values if the real class label of a port is one and it is misclassified. Besides, we put cost metric c as zero for the successfully classified cases, that are, if the real class label is zero and it is not visited, and the real class label is one, and it is visited, in other words, the diagonal elements in the cost matrix, which means that successively classified cases do not put the extra relative cost to the ideal energy consumption value.

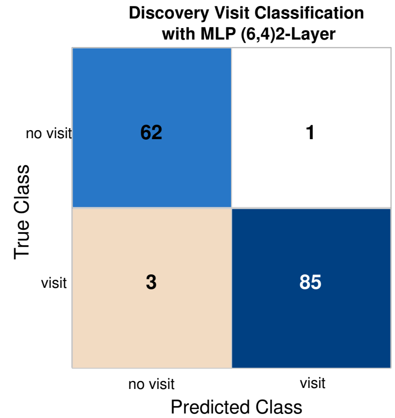
We perform relative cost analysis by using the Bayesian Decision Theory basics, and use the following formula to calculate the relative cost of DT-GDP, $Cost = \underset{iters}{\operatorname{argmax}}\{\sum_{actions} \frac{\sum_{misclass.}|c|}{E_{total}}\}$. In the cost calculation, c values are used as defined in the loss matrix.

We compare the relative cost values of DT-GDP with the TPD and TCD. According to our experimental results (given in Table VI), we see that there is 45% decrease in relative cost compared to the average relative cost of TCD and TPD. Also, we observe 23% decreased relative cost value for the simulated sparse topology.

4) Learning Method Performance Analysis: We investigate the performance results MLP. We use cross-entropy as the loss function and update the training parameters according to the minimized cross-entropy values. For example, Fig. 11a shows that the cross-entropy has its maximum value at the first



(a) Calculated cross-entropy values for each epoch



(b) Confusion matrix for MLP output

Fig. 11: MLP performance results

TABLE VII: Performance Results of Learning Methods with Different Evaluation Metrics

	MLP		Naive Bayes		Quadratic Discriminant		KNN		Decision Tree		SVM	
	Sparse	Dense	Sparse	Dense	Sparse	Dense	Sparse	Dense	Sparse	Dense	Sparse	Dense
Accuracy (%)	**97.33**	**98**	83.1	90.1	91.7	95.2	91.7	95.2	94.3	**96**	94	**96.6**
Training time (sec)	2.58	2.73	0.8421	0.97	0.87	1.05	0.87	1.05	**0.65**	**0.65**	1.87	2.03
Prediction speed (obs/sec)	∼**4850**	∼5000	∼8350	∼8500	∼6300	∼6600	∼6300	∼6600	∼**12000**	∼**12000**	∼4150	∼4500
TPR-Recall (%)	**98.0**	**96.7**	85.7	93.5	91.5	**96.4**	91.5	**98.0**	92.6	94.7	90.5	**96.0**
FPR (%)	**2.0**	**3.3**	14.3	6.5	8.5	**3.6**	8.5	**2.0**	7.4	5.3	9.5	**4.0**

epoch, and within the algorithm run, it is tried to be minimized. During the running of MLP, we achieve the best validation performance at epoch 36. Furthermore, we examine the results of MLP by forming the confusion matrix of the classification problem. According to Fig. 11b, accuracy and recall values are calculated as 98% and $\sim 95.4\%$.

TABLE VIII: One-feature vs Multi-feature effect to the MLP Prediction

	One-feature	Multi-feature
Accuracy (%)	92.6	98
TPR (%)	91.5	96.7
FPR (%)	8.5	3.3

Furthermore, we investigate the one-feature and multi-feature effects on the prediction success. We first use the port utilization feature in the form of time-series data. We perform periodic sampling of 12-hour, 24-hour, and 36-hour in order to add auxiliary port utilization features to our dataset. After we feed the (6,4) 2-layered MLP architecture with this new input data that considers only port utilization information with the minimum, maximum and average values for three different sampling periods. To measure the multi-feature effect, we feed the same MLP network with the dataset that includes all twenty-five features (given in Table III). As seen from Table VIII, a multi-feature case results in more successful prediction performance due to highly correlated features and information gain. The executive summary for all of the simulated learning methods for multi-featured dataset is given in TableVII. By looking at these results, we can deduce the following points: (i) Although MLP requires more runtime and make prediction more slowly, it achieves a significant accuracy result which is the desired performance in the newly designed discovery policy, (ii) The best recall values for the classification task is achieved with MLP for both the sparse and dense topology structures, (iii) SVM and Decision Tree models can be preferred if less runtime (training time and prediction speed) is required for the classification problem.

IV. CONCLUSION

With the tremendous interest in green network management strategies, many service providers try to develop new methodologies regarding the main network management functions, especially for network discovery. Since resource usage plays a significant role in the energy consumption in discovery, we propose a novel network discovery policy called DT-GDP that jointly uses the individual energy consumption and the visit behavior of an IP port. In addition, it integrates the digital twin technology into the network discovery service. Thanks to this integrated design, we not only achieve to reduce the total energy consumption but also the total network management cost of a service provider, making it possible to examine the network topology in a virtual environment. In DT-GDP, there are two sub-modules, one is performing a novel energy definition, and the other is predicting the visit behavior for IP ports. MLP is used in the implementation of the brain of the digital twin. Experimental results show that DT-GDP performs well in terms of the total number of visited ports, total energy consumption, percentage resource usage, and the total relative cost of discovery service.

For future work, we are planning to revise the energy formula by taking into consideration the requested traffic types. In this way, we are planning to make a distinction between the different types of traffic and their corresponding energy consumption. In addition, as another future work case, we plan to examine the effect of DT communication problems, such as synchronization, on DT-GDP to enhance our energy consumption model.

Acknowledgments

At this work, Kubra Duran is supported by DeepMind Scholarship Program and 100/2000 YÖK Ph.D. Scholarship Program.

References

[1] M. O. I. Musa, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Bounds on greentouch greenmeter network energy efficiency," *Journal of Lightwave Technology*, vol. 36, no. 23, pp. 5395–5405, 2018.

[2] U. S. A. T. W. B. International Energy Agency (IEA), IRENA and WHO, "Tracking sdg 7. the energy progress report," 2020.

[3] S. Hu, X. Chen, W. Ni, X. Wang, and E. Hossain, "Modeling and analysis of energy harvesting and smart grid-powered wireless communication networks: A contemporary survey," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 2, pp. 461–496, 2020.

[4] W. Sun, H. Zhang, R. Wang, and Y. Zhang, "Reducing offloading latency for digital twin edge networks in 6g," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 12 240–12 251, 2020.

[5] Q. Zhu, X. Yu, Y. Zhao, A. Nag, and J. Zhang, "Auxiliary-graph-based energy-efficient traffic grooming in ip-over-fixed/flex-grid optical networks," *Journal of Lightwave Technology*, vol. 39, no. 10, pp. 3011–3024, 2021.

[6] E. ZSM, "Zero touch network and service management (zsm) landscape, version 1.1. 1," *ETSI: Sophia Antipolis, France*, 2020.

[7] ITU-T, "Architectural framework for machine learning in future networks including imt-2020: Y. 3172," 2019.

[8] H. Ahmadi, A. Nag, Z. Khar, K. Sayrafian, and S. Rahardja, "Networked twins and twins of networks: An overview on the relationship between digital twins and 6g," *IEEE Communications Standards Magazine*, vol. 5, no. 4, pp. 154–160, 2021.

[9] N. P. Kuruvatti, M. A. Habibi, S. Partani, B. Han, A. Fellan, and H. D. Schotten, "Empowering 6g communication systems with digital twin technology: A comprehensive survey," *IEEE Access*, vol. 10, pp. 112 158–112 186, 2022.

[10] A. Masaracchia, V. Sharma, B. Canberk, O. A. Dobre, and T. Q. Duong, "Digital twin for 6g: Taxonomy, research challenges, and the road ahead," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 2137–2150, 2022.

[11] H. Wang, Y. Wu, G. Min, and W. Miao, "A graph neural network-based digital twin for network slicing management," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1367–1376, 2022.

[12] M. Liu, S. Fang, H. Dong, and C. Xu, "Review of digital twin about concepts, technologies, and industrial applications," *Journal of Manufacturing Systems*, vol. 58, pp. 346–361, 2021.

[13] K. Duran, B. Karanlik, and B. Canberk, "Ai-driven partial topology discovery algorithm for broadband networks," in *2021 IEEE 18th Annual Consumer Communications Networking Conference (CCNC)*, 2021, pp. 1–6.

[14] L. Nasraoui and S. Ikki, "Neighbor discovery for prose and v2x communications," *IEEE Internet of Things Journal*, vol. 8, no. 9, pp. 7241–7251, 2021.

[15] J. Ye, G. Fei, X. Zhai, and G. Hu, "Network topology inference based on subset structure fusion," *IEEE Access*, vol. 8, pp. 194 192–194 205, 2020.

[16] S.-Y. Park, C. S. Shin, D. Jeong, and H. Lee, "Dronenetx: Network reconstruction through connectivity probing and relay deployment by multiple uavs in ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 11 192–11 207, 2018.

[17] J. Liu and P. Srikantha, "Adaptive topology reconfiguration in smart distribution networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 7, pp. 716–729, 2021.

[18] Y.-C. Chang, H.-T. Lin, H.-M. Chu, and P.-C. Wang, "Efficient topology discovery for software-defined networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1375–1388, 2021.

[19] L. Ochoa-Aday, C. Cervelló-Pastor, and A. Fernández-Fernández, "etdp: Enhanced topology discovery protocol for software-defined networks," *IEEE Access*, vol. 7, pp. 23 471–23 487, 2019.

[20] Y. Jia, L. Xu, Y. Yang, and X. Zhang, "Lightweight automatic discovery protocol for openflow-based software defined networking," *IEEE Communications Letters*, vol. 24, no. 2, pp. 312–315, 2020.

[21] Z. Shen, H. Jiang, Q. Dong, and B. Wang, "Energy-efficient neighbor discovery for the internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 684–698, 2020.

[22] Y. Liu, H. Wang, L. Cai, X. Shen, and R. Zhao, "Fundamentals and advancements of topology discovery in underwater acoustic sensor networks: A review," *IEEE Sensors Journal*, vol. 21, no. 19, pp. 21 159–21 174, 2021.

[23] E. Ak, K. Duran, O. A. Dobre, T. Q. Duong, and B. Canberk, "T6conf: Digital twin networking framework for ipv6-enabled net-zero smart cities," *IEEE Communications Magazine*, pp. 1–7, 2023.

Kübra Duran received her BSc degree in Electrical and Electronics Engineering from Osmangazi University in 2016 and her MSc degree in Computer Engineering from Istanbul Technical University in 2021. She has awarded with the Best Theoretical MSc Thesis Award'22 by IEEE Communications Society Turkey Chapter, and the Second Best MSc Thesis Award'22 by IEEE Computer Society Turkey Chapter. She is currently a PhD candidate in Computer Engineering Department of Istanbul Technical University. Her research interests include Digital Twin Networks, Intelligent and Green Next Generation Network Management Systems.

Berk Canberk (SM'16, M'03): Dr. Berk Canberk is a Professor in the School of Computing, Engineering and The Built Environment, where he leads interdisciplinary research and initiatives in AI-enabled Digital Twins, IoT Communication, and Smart Wireless Networks. He is an IEEE Senior Member, and he acts as an active Associate Editor at several world-leading academic journals such as IEEE Transactions on Vehicular Technology (Scopus Q1 Quartile) since 2016, Elsevier Computer Networks Journal (Scopus Q1 Quartile) since 2013, Elsevier Communication Networks Journal (Scopus Q1 Quartile) since 2017, and IEEE Communications Letters (Scopus Q1 Quartile) between 2018-2021. He's actively involved in several conferences as TPC chair and Organizing Committee Member. He's been an Adjunct Professor within the Department of Electrical and Computer Engineering at Northeastern University USA since 2017, and Adjunct Faculty within the Department of Artificial Intelligence and Data Engineering at Istanbul Technical University (ITU) since 2022. He was an Associate Professor at Department of Computer Engineering at ITU between 2016-2021, and full Professor between 2021-2022.