electronics

MDPI

Article

Network Traffic Prediction Model in a Data-Driven Digital Twin Network Architecture

Hyeju Shin, Seungmin Oh, Abubakar Isah, Ibrahim Aliyu, Jaehyung Park and Jinsul Kim *

Department of ICT Convergence System Engineering, Chonnam National University, Gwangju 61186, Republic of Korea; sinhye102@gmail.com (H.S.); osm5252kr@gmail.com (S.O.); abubakarisah@jnu.ac.kr (A.I.); aliyu@jnu.ac.kr (I.A.); hyeoung@jnu.ac.kr (J.P.)
* Correspondence: jsworld@jnu.ac.kr

Abstract: The proliferation of immersive services, including virtual reality/augmented reality, holographic content, and the metaverse, has led to an increase in the complexity of communication networks, and consequently, the complexity of network management. Recently, digital twin network technology, which applies digital twin technology to the field of communication networks, has been predicted to be an effective means of managing complex modern networks. In this paper, a digital twin network data pipeline architecture is proposed that demonstrates an integrated structure for flow within the digital twin network and network modeling from a data perspective. In addition, a network traffic modeling technique using data feature extraction techniques is proposed to realize the digital twin network, which requires the use of massive streaming data. The proposed method utilizes the data generated in the OMNeT++ environment and verifies that the learning time is reduced by approximately 25% depending on the feature extraction interval, while the accuracy remains similar.

Keywords: digital twin network; artificial intelligence; traffic prediction; graph neural network; data pipeline; data processing

Citation: Shin, H.; Oh, S.; Isah, A.; Aliyu, I.; Park, J.; Kim, J. Network Traffic Prediction Model in a Data-Driven Digital Twin Network Architecture. *Electronics* **2023**, *12*, 3957. https://doi.org/10.3390/electronics12183957

Academic Editor: Christos J. Bouras

Received: 31 July 2023
Revised: 11 September 2023
Accepted: 14 September 2023
Published: 20 September 2023

1. Introduction

With the emergence of immersive services, including virtual reality/augmented reality, hologram content, and the metaverse, communication networks are becoming increasingly complex, thereby enhancing the complexity of network management [1]. The digital twin network (DTN) or the network digital twin (NDT), a technology that integrates the digital twin (DT) with communication networks, has emerged as a promising new intelligent network management technology to efficiently handle these sophisticated networks [2,3]. The role of DT technology has grown in importance across a variety of fields in parallel with the advancement of simulation and computing technology [4]. There is active research in digital cities, including smart cities [5–7], and intelligent manufacturing technologies, including smart factories [8–10]. DT technology, which links physical space with virtual digital space, enables real-time data management based on physical models along with the full lifecycle operation and maintenance of data. This can be achieved by integrating a multi-variable, multi-scale, and multi-probability based computer simulation process to generate a virtual twin entity corresponding to the physical entity. Furthermore, it can mimic all states of the physical entity in real time, providing high fidelity and high integration. This enables the monitoring of the entity's condition throughout the lifecycle process [11]. In the DTN, physical entities represent actual physical network systems that exist in the real world. These entities can be mapped to virtual twin networks in the digital domain to facilitate the comprehensive management of complex networks.

In this context, accurate modeling of the real environment in virtual reality has a significant impact on the performance of the DTN. Accordingly, various researches [12–14] on modeling traffic flow for accurate simulation, standardization, and research on DTN architecture have been conducted together [2,15–19]. However, the existing architectures

are abstracted, making it difficult to confirm the specific data flows for the actual DTN implementation. According to [20], the data collection, storage, processing, and provisioning processes based on the data types should be provided for communication network management. In addition, according to [21], the network state data can be classified by role, and appropriate data-processing processes are required for DTN configuration. To the best of our knowledge, there has been no research that provides an integrated structure for the periodic data flow within the DTN. In addition, there is no discussion on the network modeling method in terms of data considering the characteristics of data in the DTN environment. Therefore, in this research, we describe an integrated architecture for the flow of data within the DTN and the processes provided for network management, with the aim to demonstrate this along with scenarios for the network traffic modeling methods for the actual DTN implementation. The proposed network traffic modeling method enables the training of the DTN artificial intelligence (AI) model, which requires extensive streaming data. This method reflects the spatial characteristics of topology data and performs a comparison with the existing methods in terms of learning time and convergence speed using an AI model that can be generalized to the scale of the topology to ensure scalability.

In this context, the main contributions of this paper are as follows:

- A data-driven DTN data pipeline architecture is proposed that encompasses the DTN lifecycle flow and processing process from a data perspective.
- An AI model learning process is proposed that combines data feature extraction techniques to enable the modeling of massive streaming data-driven DTN network traffic data.
- By combining the proposed network traffic modeling technique with a spatio-temporal feature compression model, a reduction of approximately 25% in learning speed compared to the existing method is confirmed while maintaining similar accuracy.

The structure of this paper is as follows. Section 2 provides a brief overview of the background of the proposed work by reviewing studies related to software-defined networks, the DTN, network traffic prediction, and the graph neural network (GNN). In Section 3, we provide a general overview of the proposed data-driven DTN data pipeline and traffic prediction model. In Section 4, the experiments and results of the traffic prediction model designed using compressed data are presented. Finally, the conclusion is given in Section 5.

2. Related Works

This section introduces several concepts that form the basis of the proposed architecture. First, a brief definition of software-defined networking (SDN) is presented, along with a discussion of its basic architecture. Next, the conceptual background and key components of the DTN are outlined. These concepts provide a foundation for the data pipeline of the data-driven DTN architecture.

2.1. Software-Defined Networks

SDN refers to the technology that separates the hardware and software functions of network equipment [22]. Through this separation, the hardware realm assumes the role of the data plane, transmitting the data. The software realm is divided into the control and application planes. As shown in Figure 1, the control plane can be considered to be the area where network policies are applied. It can be logically centralized and programmed via an SDN controller. This provides flexibility in network control, flow management, and congestion control. The application plane functions as the user support plane and performs policy implementation tasks.

Traditional network equipment suffers from a strong hardware dependency. This leads to difficulties in integrated management due to closed designs. An example of this issue would be the use of different custom semiconductors by different equipment vendors. SDN, on the other hand, keeps only the data transmission processing within the network infrastructure and delegates the remaining functions such as configuration, access control lists, and provisioning to a controller. This approach enables the network to be

operated easily and quickly and provides flexible and efficient network management [23]. It combines all pieces of the network equipment into an intelligent system and solves the scalability problems intrinsic to traditional centralized structures.

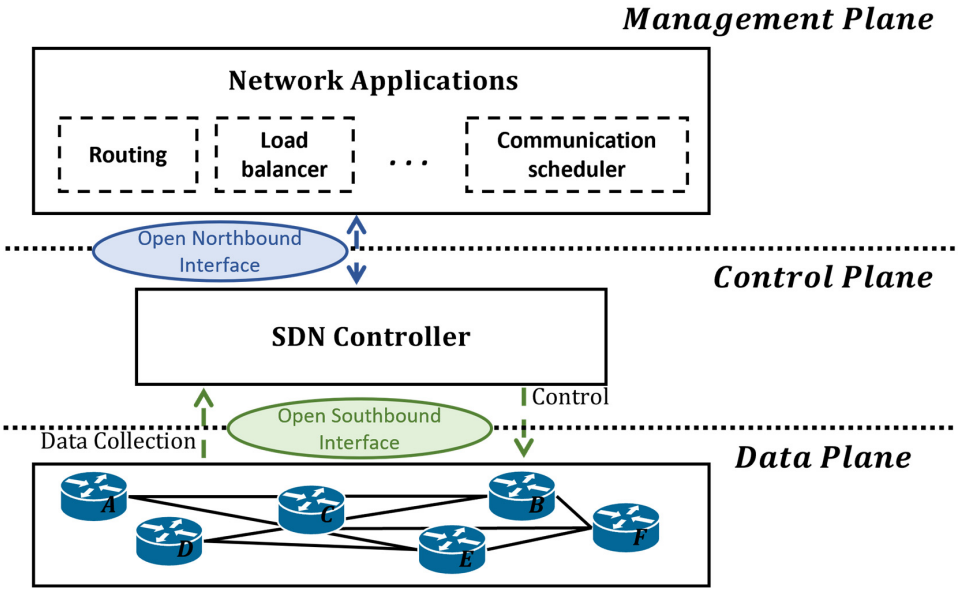

Figure 1. SDN architecture overview.

SDN is an innovative paradigm in networking and offers on-demand resource allocation, easy reconfiguration, and programmable network management. The separation of the control and data planes in the network enables the flexible and consistent execution of network management and control [24]. As a result, the adjustment of the allocated bandwidth and paths for user services becomes dynamically manageable, facilitating traffic control and management. To this end, the SDN control plane collects real-time network status information from the data plane. It can provide a path that meets quality-of-service requirements between communicating endpoints according to user requirements.

The creation of intelligent networks has become achievable via the abstracted central controller. Network intelligence is enhanced via programmable software applications at the network control plane and the integration of AI. Recently, the emergence of the DTN and network management technology has facilitated the discussion of network automation technologies based on the SDN structure.

2.2. Digital Twin Network

The DTN refers to a technology that efficiently analyzes, diagnoses, simulates, and controls physical networks using the data and models within a virtual digital network that is constructed using DT technology across various network management systems and applications [15,16]. According to this definition, four key elements are required for a DTN: data, model, mapping, and interface.

The data are the foundation for building a DTN, and building an integrated data-sharing warehouse that acts as a single source of truth for the DTN facilitates the efficient storage of physical network configurations, topology, operational status data, logs, user business records, and real-time data, thereby providing data services to the network twin. The model acts as a functional source for the DTN, creating a variety of model instances via flexible combinations to provide different network applications. Mapping is required to provide a high-fidelity visualization of the physical network entities using the virtual twin network. This differentiates the DTN from a network simulation system and allows it to accurately model the state and behavior of physical network entities. The interface is a key technology for achieving physical–virtual synchronization. It connects the network service applications and physical network entities using standardized interfaces which collect and

control real-time information about the physical network. This facilitates timely diagnosis and analysis. Physical–virtual synchronization is the process of updating a virtual twin entity based on the state of a physical entity. This can be achieved by collecting real-time data using an interface. Additionally, the optimized results derived from the DTN can be distributed to the physical network and controlled using the interface.

A twin network built on these four elements provides the analysis, diagnosis, simulation, and control of the physical network throughout its entire lifecycle using optimization algorithms, management methods, and expert knowledge. In the Internet Research Task Force (IRTF), the reference architecture of the DTN is presented as a structure that includes three layers and three domains, as shown in Figure 2, according to the definition of these key elements [15]. The three layers are the physical network layer, DTN layer, and network application layer. The three domains within the DTN layer represent the data domain, model domain, and management domain, which correspond to the data repository, service mapping model, and DTN management module subsystems, respectively. Here, the subsystem refers to one of the elements that make up the system, and it also means that it is a system in itself.

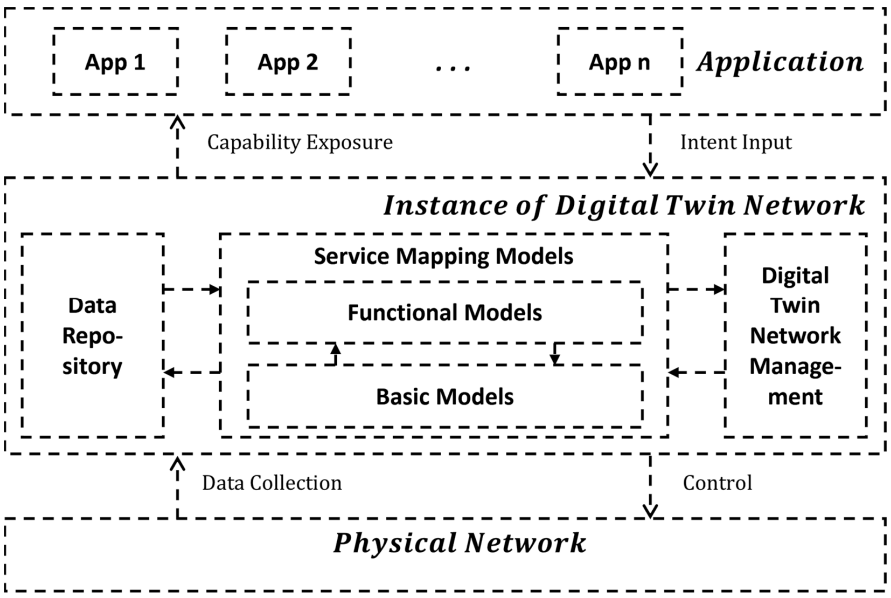

Figure 2. DTN reference architecture.

The physical network layer consists of physical network equipment that exchanges the network data and control information with the DTN layer via the southbound interface (SBI). This layer can consist of various types of physical networks, including mobile access networks, transport networks, mobile core networks, backbone networks, data centers, enterprise networks, and the industrial Internet of Things. The DTN layer consists of three subsystems: a data repository, a service mapping model, and DTN management. The data repository collects and stores a wide range of network data and provides an integrated interface to the service mapping model, facilitating the mapping of data-to-data services and models. The service mapping model completes the data-based modeling and provides data model instances for different network applications, maximizing the flexibility and programmability of network services. The network application layer communicates requirements to the twin network layer via the northbound interface (NBI) and provides services to the twin network layer using the modeled instances. After extensive validation, the twin network layer pushes control updates to the physical network equipment via the SBI.

The use of such a DTN technology enables the rapid deployment of innovative network technologies and a wide range of applications such as network operation, maintenance, optimization, network visualization, intent verification, and network self-tuning devices.

This can be achieved with lower cost, higher efficiency, and greater stability compared to traditional network services. To effectively use such a DTN, the accurate modeling of the actual network as a virtual twin network must be ensured [25], and research is being conducted to model the future state of communication networks and data using AI prediction models [26–29].

2.3. Network Traffic Prediction

Network traffic prediction is becoming increasingly complex and diverse and is considered very important for network operation and management [30]. This is because predicting future traffic conditions (e.g., delay, traffic volume, etc.) in advance can improve the network performance. In addition, it can be used as a network analysis model to evaluate network policies before deploying network settings in the physical network within the closed loop between the DTN physical layer and the twin layer [12] and update the network optimization model accordingly.

Various studies based on recurrent neural network (RNN) models, which can learn temporal patterns and long-range dependencies from large-scale sequences of arbitrary length, have been conducted for traffic prediction tasks [31,32]. Recently, various studies have also been conducted using GNN models that can reflect the spatial characteristics of communication networks to improve accuracy [12,33–35]. However, discussions are still ongoing regarding approaches for learning large amounts of the data generated in real time in large-scale networks [15,16].

2.4. Graph Neural Network

Unlike a convolutional neural network (CNN) and RNN, a GNN takes non-Euclidean spatial data such as the network topology, molecular structure, and knowledge graphs as the input. CNN, which is widely used for feature extraction from two-dimensional data, has also been used for feature extraction in the network space. However, it has limitations in processing the matrix data instead of the graph data; therefore, it does not perform well in predicting traffic flow. Accordingly, the graph convolutional network was developed as a modification to effectively capture features from the map data. In addition, several studies have recently been conducted using GNN on the input data with a graph structure to reflect the spatial characteristics of network topology. In [33], a GNN-based autoencoder model called the gated graph autoencoder network is proposed to accurately predict the delay of SDN networks. In [34], RouteNet, which uses GNN models for network modeling and optimization, is proposed; the GNN-based RouteNet can predict key performance indicators such as average delay and jitter for each network source/destination. Active discussions on the use of GNN models to reflect actual network characteristics are also ongoing in the standardization efforts for DTN construction [35].

2.5. Long Short-Term Memory

Long Short-Term Memory (LSTM) is a neural network structure designed to overcome the vanishing gradient problem that occurs in traditional RNNs when dealing with information that is far from the output. Despite the existence of many efficient models for predicting time series data, LSTM has proven to be one of the most promising models for solving time series prediction problems [36]. LSTM can store past information in the network, allowing it to remember long-term information. In this way, future results are influenced by past input values, and it is widely used in various time series prediction studies [37–39]. It is also widely used for predicting the performance data of communication networks that change over time [40–43].

3. System Overview

The goal of this section is to present approaches from a data perspective for modeling traffic data in the DTN. First, we propose a data-driven DTN-customized data pipeline archi-

tecture and explain the process of training useful data-based AI models using lightweight data feature extraction techniques that can be incorporated into this architecture.

3.1. Reference Architecture

Consider the reference DTN architecture based on the existing literature [15,16,44–47] as shown in Figure 3. The lowest layer can target the physical networks belonging to the real world (e.g., mobile access networks, mobile core networks, etc.). Network equipment entities within such networks are interconnected and connected to other entities via connectivity features provided by the network infrastructure.

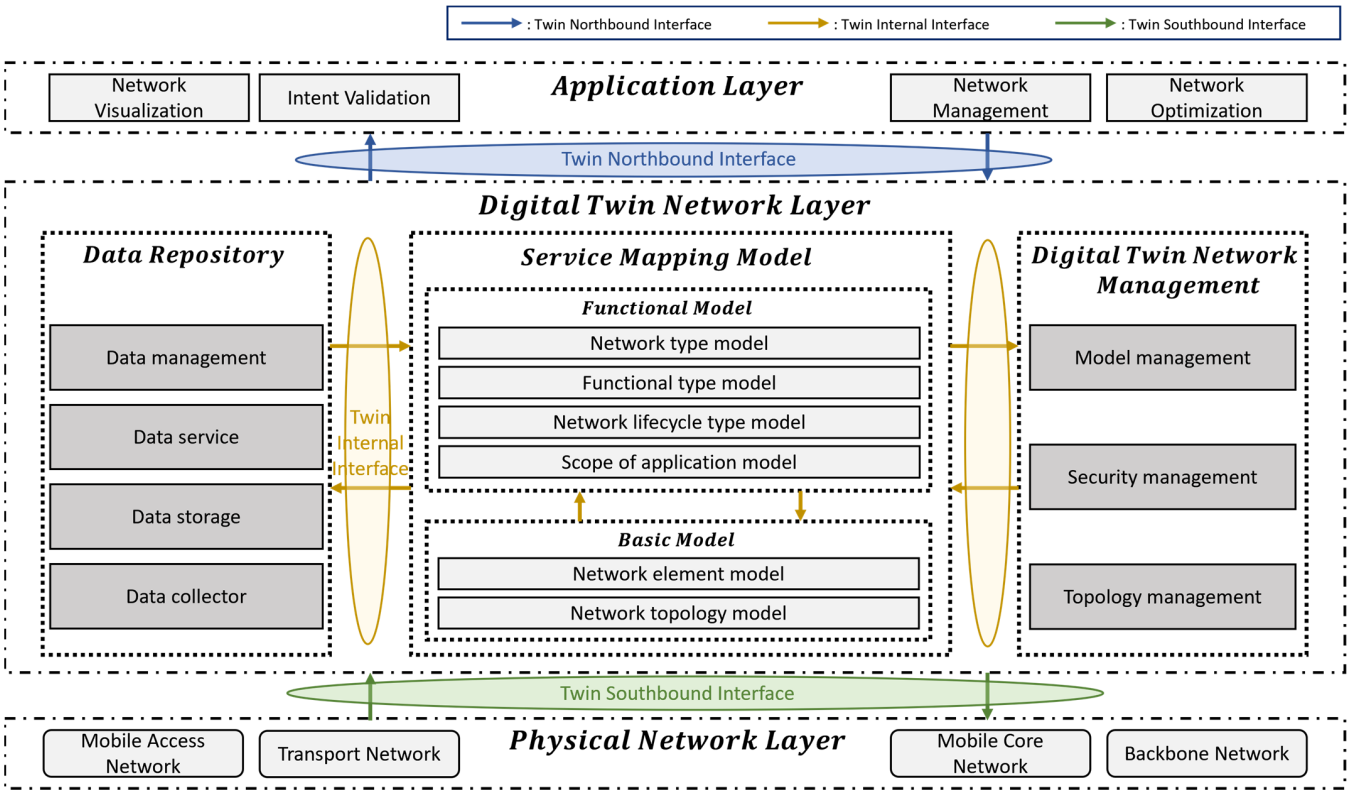

Figure 3. Reference architecture of a DTN.

The middle DTN layer is a virtual layer where the digital representations of network entities located in the physical network layer are placed. The data repository, a subsystem of the DTN layer, includes a data collector, data storage, data services, and data management as its subsystems. Network entities are typically collected by the data collectors located in the data repositories, and after undergoing a certain process, they are stored in the repository or provided as a service mapping model by the data management and data service systems. The data provided to the service mapping model is used to build a model for performing user-intent-based tasks. AI can be used in this process, and the trained AI model can be stored in the model database as unstructured data.

The basic models within the service mapping model of the DTN layer are classified into network element models and network topology models. Functional models can be built, including network-type models, functional-type models, network lifecycle-type models, and scope-of-application models [46]. These sub-models receive data from the data repository and are combined into the service mapping model based on the user intent by the DT management.

The DT management system of the DTN layer can consist of a topology management system, model management system, and security management system. The topology management system manages the digitized network topology and can create a virtual

twin topology. The model management system provides overall support to ensure that the models in the DTN perform their tasks appropriately according to user intent and supports model instance creation, storage, update, model combination management, and model-application software connection management. The security management system ensures the security of the data, models, and interaction data throughout the lifecycle using authentication management, access management, and encryption.

In the top application layer, a user interface such as network visualization is provided. This allows users to manage the network using the DTN. It communicates user intent to the DTN via intent validation and provides functionalities such as network management and optimization, enabling users to manage the network efficiently.

3.2. Proposed Data-Driven DTN Data Pipeline Architecture for Network Management

Various structural reference architectures have been proposed for the effective use of the DTN. However, the existing structures do not provide concrete insights into the overall data flow, including the process of providing data to the AI model within the DTN [2,15–19]. Moreover, it is essential to understand the overall process of providing customized data for various scenarios within the DTN and the intelligent model for effective network management. Therefore, this section presents a data pipeline architecture that can help understand the overall flow and processing of the DTN from a data perspective before explaining the AI model. The data-driven DTN data pipeline architecture is shown in Figure 4.

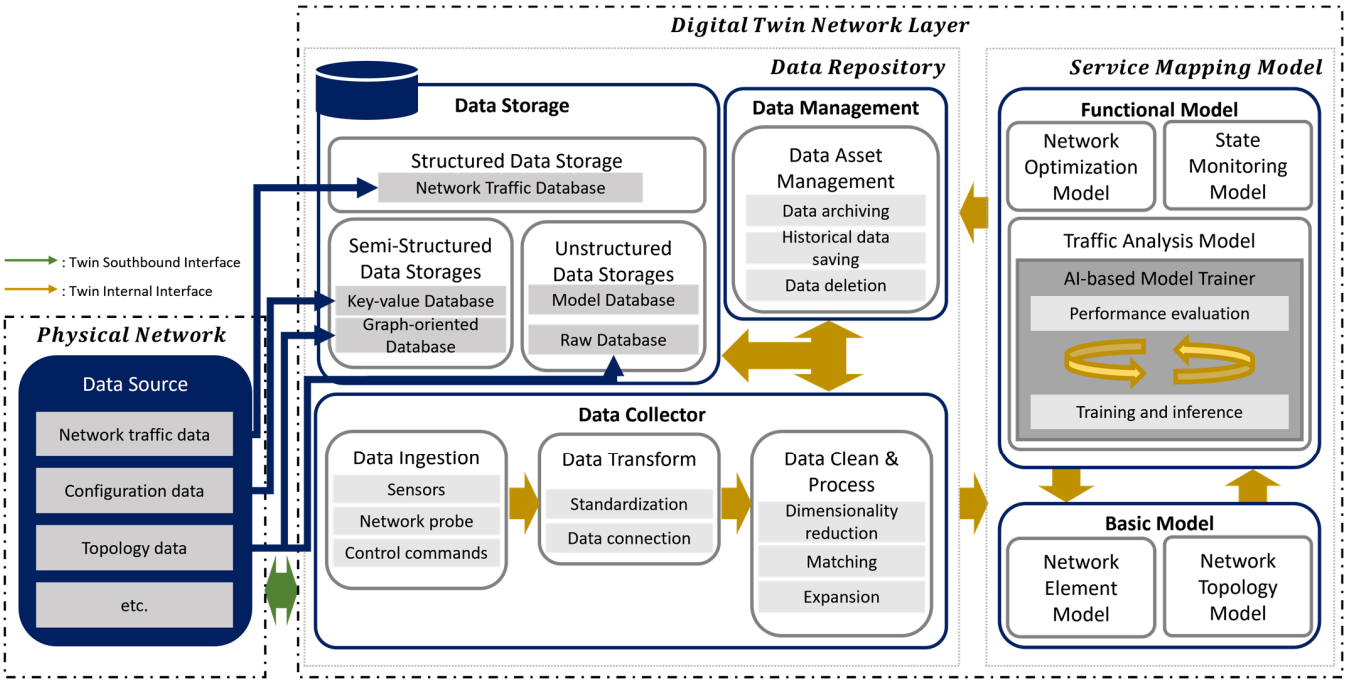

Figure 4. Proposed data-driven DTN data pipeline architecture.

The data that can be collected from the physical network on the far left for network management include the network equipment data, network traffic data, and network performance data. The network equipment data include the geometric data, status data, event data, and topology data [44]. The data can be classified according to their form into types such as the log/event type, metric data, flow data, packet data, configuration data, and forwarding and routing data [16]. The data are stored in an appropriate database according to the structured information as either structured, semi-structured, or unstructured. For example, the topology data collected from the physical layer can be stored as unstructured raw data in their unprocessed state. After collection, they can be processed into a graph form by preprocessing and then stored in a semi-structured graph-oriented database.

The data are collected via a data collector, where they undergo ingestion, transformation, cleaning, and processing stages [48]. After being ingested into the collector via devices such as sensors, they are transformed using normalization and data linkage. Then, they undergo the data cleaning and processing stages including dimensionality reduction and expansion. These pre-processed data are sent either to the data storage or to the service mapping model via the data management system, depending on the user intent.

The service mapping model that receives the data can support network management according to the network scenarios using various models. For example, models categorized as function-type models include a status-monitoring model, traffic analysis model, fault diagnosis model, and security drill model. Further, the models categorized as network lifecycle-type models include a network planning model, network configuration model, network optimization model, and network operation model. In this case, AI models can be used to build each model, and the service mapping model portion of Figure 5 represents a learning model for the traffic analysis model.

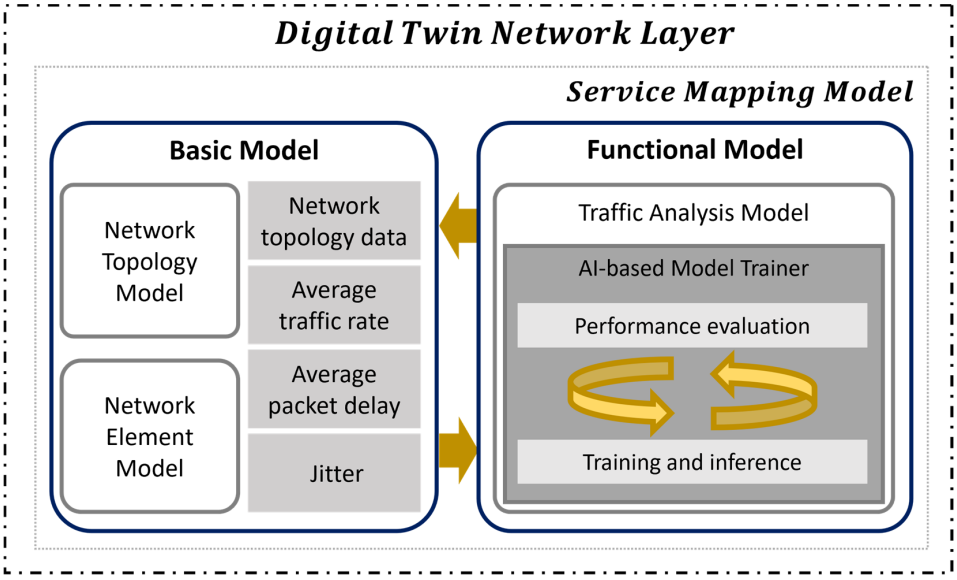

Figure 5. Overview of the proposed network traffic data prediction method.

The data that have passed through the physical network layer and the DTN layer are delivered to the user in the form of AI for IT operations (AIOps), applications, or network management analytic tools at the application layer.

3.3. Proposed Network Traffic Data Modeling Method

This section explains the process of training an AI model with useful data using feature extraction techniques on the data structured appropriately for the scenario, as shown in Figure 5. Feature extraction refers to using only useful data when representing a large amount of data [49]. The analysis of streaming the data generated in real time requires high computing power and a large amount of memory, and it takes a long time to train. In addition, there may be a problem of overfitting the AI model to the initial training data. Therefore, selecting only meaningful data from the original data can maintain accuracy while reducing training time by predicting traffic based on the selected input data.

In the proposed network traffic data modeling method, the input data are prepared by extracting data features according to the K time steps from the original data. The prepared data are then used as the input to the AI model, which combines traditional time-series data prediction algorithms after passing through a GNN algorithm.

The traditional method employing the original data uses the $(T_n - t, T_n - 1)$ data to derive the T_n values using a prediction process. The length of the training data is $len(T_n)$. The combined method of the proposed data feature extraction process uses T'_n data to

derive T_n values. In this case, the length of the training data is $len(T'_n) = \frac{len(T_n)}{K}$, where K is a scalar for the key data extraction interval. In the proposed network data-based modeling, only the key data are extracted and used for the transfer and learning to comprehensively reflect the big data occurring in real time. Typical key data includes the start point, end point, and maximum and minimum values.

The learning process of the AI model for network traffic modeling is illustrated in Figure 5. When the topology data representing the adjacency relationship between nodes in a communication network are fed to a GNN, the spatial characteristics of the network can be compressed according to connectivity. Then, the network performance data generated over time can be compressed using an RNN model to capture the temporal characteristics of the data, allowing the prediction of new traffic data based on the learned features. The overall learning process of this model is as follows: (1) Spatial features are compressed using a graph CNN with adjacency matrices composed of topology data. (2) Meaningful data are selected via the data feature extraction tasks, and the size of the data is reduced by $\frac{4}{K}(K > 4)$ times. (3) The selected data form the input of a one-dimensional RNN via the flatten and dropout layers. (4) Each data element passes through a long short-term memory network to learn the temporal features and is output as one-dimensional feature data. (5) Finally, the predicted results are obtained by feeding the data to a dense layer. Through this process, the state of the network traffic data can be modeled, and the future state values can be predicted.

The LSTM network used at this time includes a forget gate, an input gate, and an output gate. The forget gate is used to forget unimportant past information and outputs a value between 0 and 1 based on the importance of the information using the sigmoid function, as shown in Equation (1).

$$f_t = \sigma\Big(W_f \cdot [h_{t-1}, x_t] + b_f\Big), \tag{1}$$

The input gate is a gate that is used to remember current information. The value obtained via the sigmoid function of Equation (2) and the result of the hyperbolic tangent go through the Hadamard product operation. This is represented as Equation (3) and outputs values between -1 and 1, as shown below.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{2}$$

$$C_t = f_t * C_{t-1} + i_t * tanh(W_C * [h_{t-1}, x_t] + b_c), \tag{3}$$

The output gate is a gate for the final result h_t, and it performs a Hadamard product operation on the hyperbolic tangent of the cell state, as shown in Equation (5). This value is output as the final result of LSTM, as shown below.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{4}$$

$$h_t = o_t * tanh(C_t), \tag{5}$$

Through this process, the future network state can be modeled and predicted based on the current state of the target network, enabling the effective management of the physical network in the digital domain

4. Experiment and Results

This section discusses the performance of the proposed AI model. The datasets used for training are shown in Figure 6. A detailed description of the performance metrics considered for the experiment is also provided in this section. The proposed model was trained in the Anaconda Jupyter Notebook on an Intel(R) Core(TM) i9-10900X CPU running at 3.70 GHz with 128 GB of RAM.

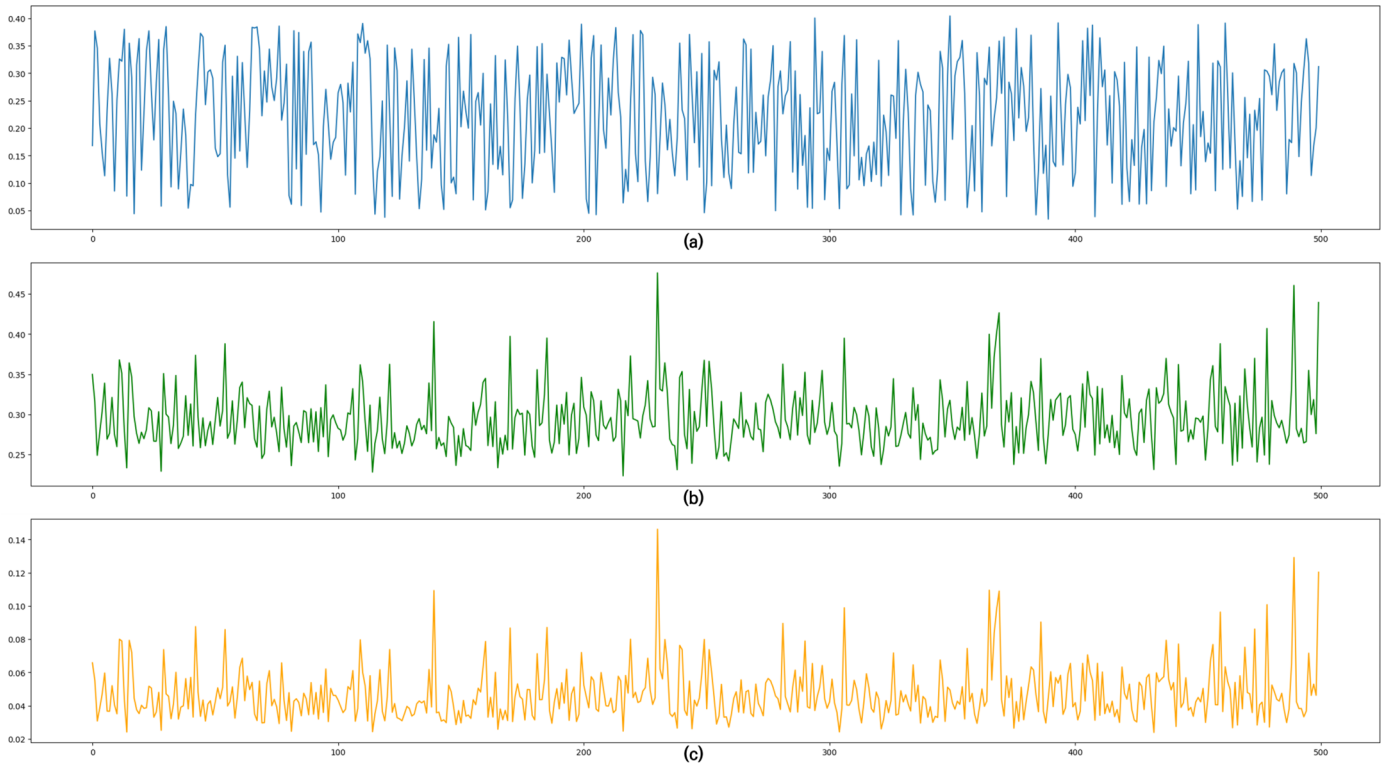

Figure 6. Examples of (**a**) average traffic rate, (**b**) average delay, and (**c**) variance of the per-packet delay in each source–destination pair.

4.1. Datasets for Network Flow Measurements

In this paper, we conducted experiments using Gigabit European Academic Networking Technology (GEANT2) data comprising 24 nodes and 37 edges as the topology data. For the network flow measurements and performance, we used the average traffic rate, average packet delay, and variance of the per-packet delay (jitter) over the packets transmitted in each source–destination pair. These were generated in the GEANT2 topology. The dataset consists of samples generated within the OMNeT++ environment. It was stochastically generated following the GEANT2 topology and according to the preset traffic intensity [50]. Further, we used root mean square propagation (RMSProp) as the optimizer and the mean squared error as the loss function to construct the neural network model. Figure 6 shows examples of the average traffic rate data, average packet delay data, and jitter data in graphical form.

4.2. Learning Results of the AI-Based Network Flow Data Prediction Model

The dataset discussed in Section 4.1 was used to train a model for predicting the performance of the network data. Figure 7 shows the loss function after applying the proposed GNN and time-series prediction algorithms to the proposed network flow measurement data. This training step required 1136 s to complete.

Figures 8–11 show the loss function according to the data feature extraction interval of the algorithm. They combine the GNN and LSTM algorithms with the proposed key data-extraction method based on the network flow measurement data. In this case, K is the time step that serves as the criterion for extracting features from the data. For a batch size of 64, the training times were 828, 541, 354, and 277 s for the K values of 5, 10, 15, and 20, respectively. For a batch size of 32, the training times were 420, 321, 291, and 203 s for the K values of 5, 10, 15, and 20, respectively.

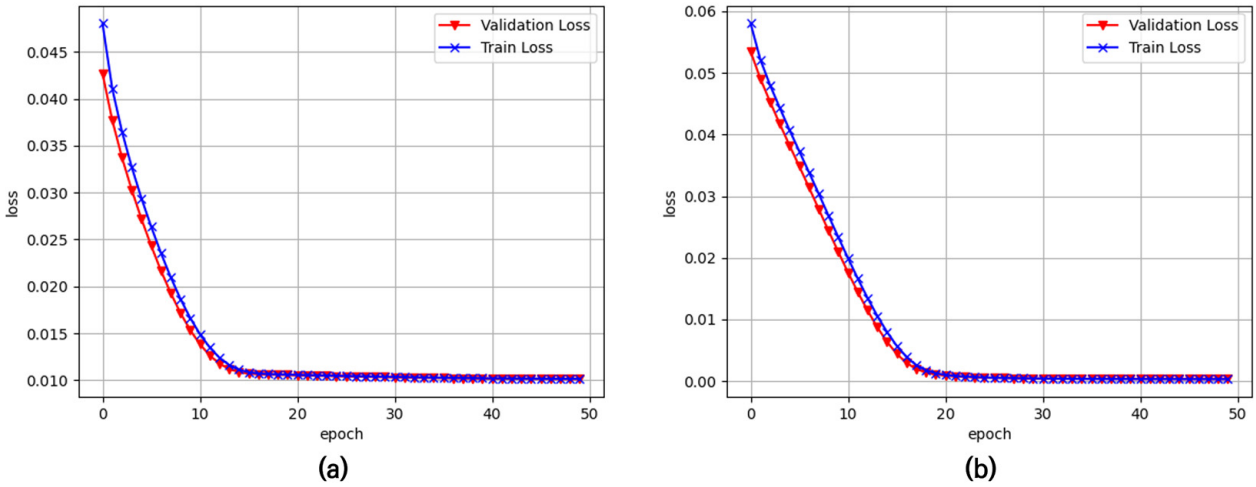

Figure 7. Loss function of the proposed prediction model with origin data with a batch size of (**a**) 64 and (**b**) 32.

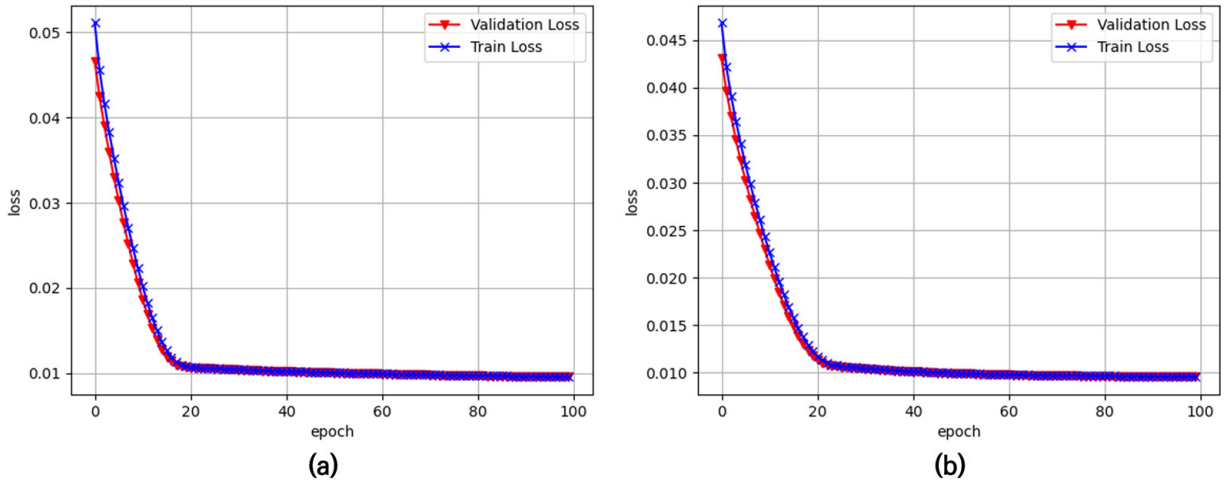

Figure 8. Loss function of the proposed prediction model for K = 5 with a batch size of (**a**) 64 and (**b**) 32.

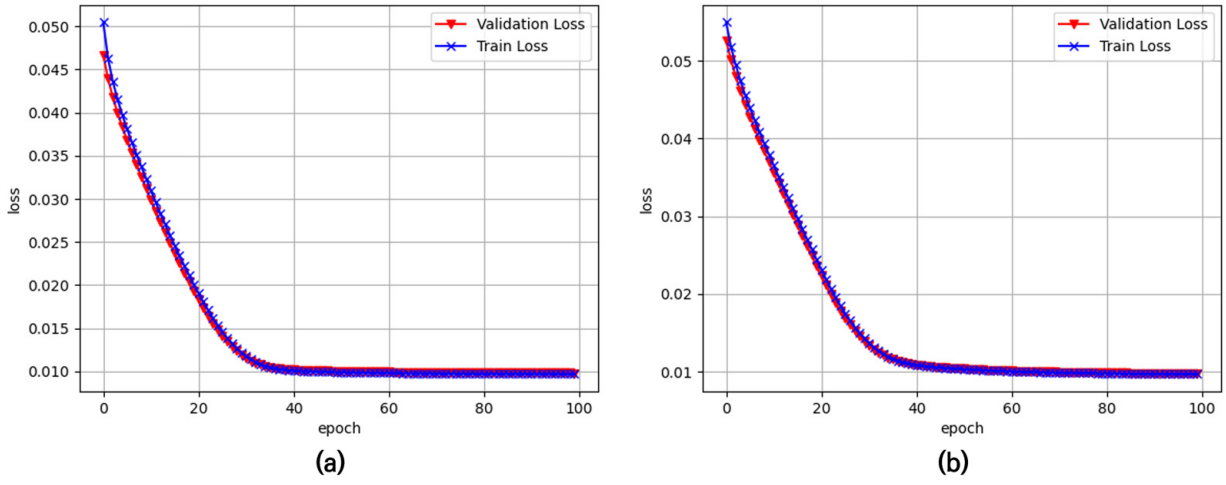

Figure 9. Loss function of the proposed prediction model for K = 10 with a batch size of (**a**) 64 and (**b**) 32.

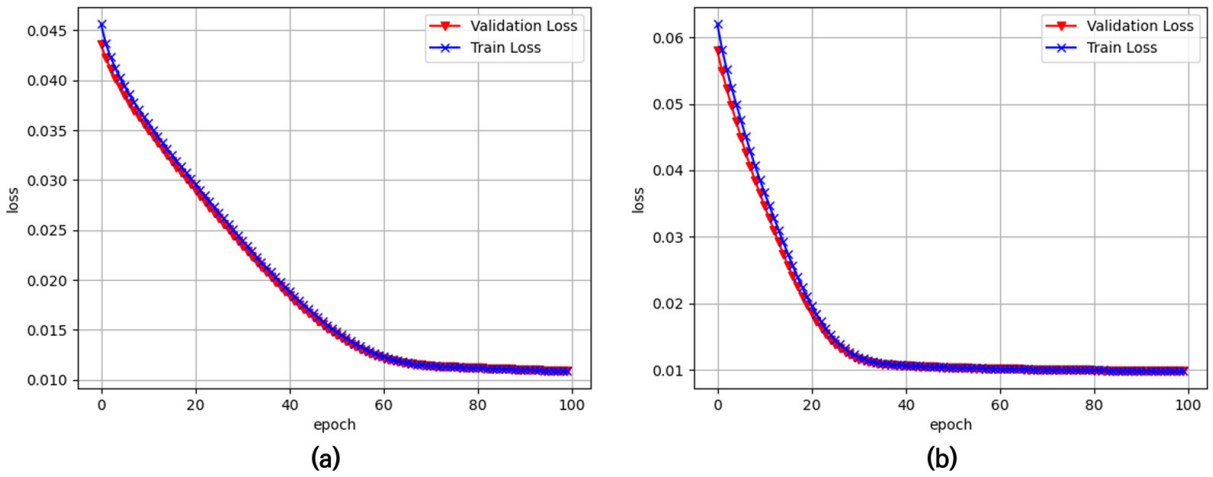

Figure 10. Loss function of the proposed prediction model for K = 15 with a batch size of (**a**) 64 and (**b**) 32.

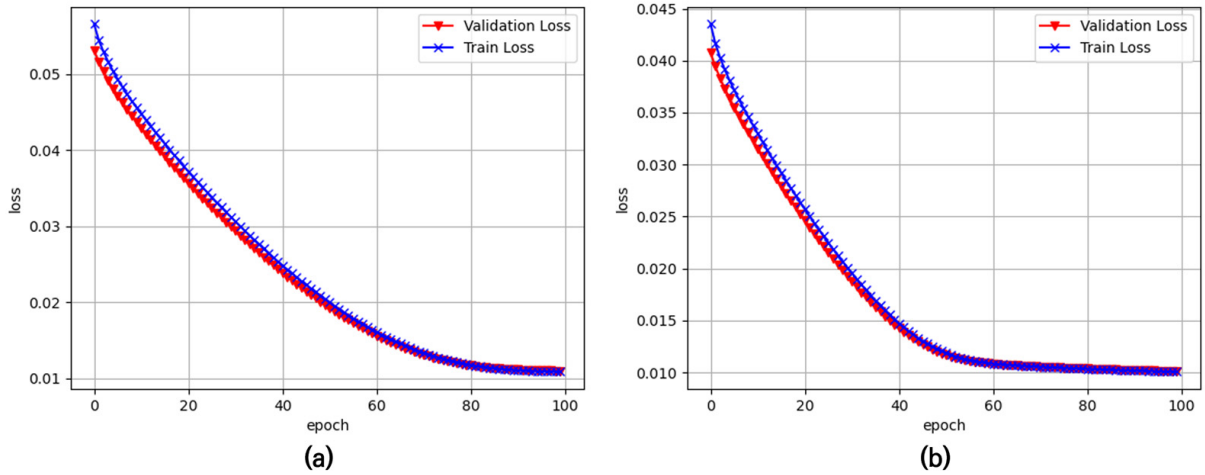

Figure 11. Loss function of the proposed prediction model for K = 20 with a batch size of (**a**) 64 and (**b**) 32.

Table 1 shows a comparative analysis of the training time of the proposed prediction model, where the batch size is varied between 32 and 64, while the criterion of data feature extraction is also varied, starting from no data feature extraction (original data) up to K = 5, 10, 15, and 20. K is the time step that serves as the criterion for extracting features from the data. In the proposed network traffic data modeling method, the input data are prepared by extracting data features according to the K time steps from the original data. As the K increases, the learning time decreases and the accuracy decreases slightly, but it can be confirmed that they are almost similar. In addition, it can be observed from the loss function in Figures 7–11 that all cases converge before 100 epochs.

Table 1. Predictive model average training time.

Data	(Batch Size = 64)			(Batch Size = 32)		
	Learning Time	**MAE**	**RMSE**	**Learning Time**	**MAE**	**RMSE**
original data	1136 (s)	0.0104	0.0182	1062 (s)	0.0098	0.0177
feature extraction (K = 5)	828 (s)	0.0107	0.0188	420 (s)	0.0104	0.0186
feature extraction (K = 10)	541 (s)	0.0132	0.0225	321 (s)	0.0131	0.0220
feature extraction (K = 15)	354 (s)	0.0169	0.0457	291 (s)	0.0156	0.0461
feature extraction (K = 20)	277 (s)	0.0167	0.0272	203 (s)	0.0184	0.0266

5. Conclusions

This paper proposes a solution from a data perspective for modeling traffic data within the DTN. There has been no specific architectural research explaining the overall data flow within the DTN or a specific discussion on improving the learning speed of AI models for modeling traffic data within the DTN based on it. Therefore, an AI model learning process based on the data feature extraction technique is proposed, which considers the learning speed for mounting on the DTN; in addition, a data-driven DTN suitable data pipeline architecture is presented for this purpose. The proposed learning method based on the data feature extraction technique is designed from a data perspective to consider the learning speed of AI models by selecting only meaningful data to reduce the amount of data, and thus, can be mounted on the DTN by considering the learning speed. To explain this, a concrete DTN architecture is presented, including the flow process of data within the DTN, such as the flow according to the data type and role classification and the process of providing data to the AI model located in the service mapping model. The proposed network traffic modeling method utilizes the data generated in the OMNeT++ environment and verifies that the learning time is reduced by approximately 25% depending on the feature extraction interval, while the accuracy remains similar. This method will help improve the speed of AI model training within the DTN by leveraging large amounts of streaming data. However, standardization work for the requirements of the DTN architecture and its internal modules is still ongoing. In the future, it seems necessary to conduct research on AI models segmented by each layer and module within the DTN.

This study sets the network analysis model within the DTN layer as the target and proposes a methodology that can be quickly trained with the real-time collected data to fit this model. However, since the data is collected in real time, the security of the data containing personal information must also be considered. Therefore, in future research, we intend to explore the application of a federated learning structure that allows AI models to be trained on distributed terminals without sharing the data. The proposed data-driven DTN architecture is expected to contribute to the implementation of intelligent network management systems by presenting a concrete architecture for the DTN design from a data perspective.

Author Contributions: Conceptualization, H.S. and J.K.; Methodology, H.S. and I.A.; Formal Analysis, S.O.; Investigation, H.S. and A.I.; Project administration, J.K.; Supervision, J.K.; Writing—original draft, H.S. and S.O.; Writing—review and editing, J.P. and J.K.; Funding Acquisition, J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Electronics and Telecommunications Research Institute (ETRI) grant funded by the ICT R&D program of MSIT/IITP [2019-0-00260, Hyper-Connected Common Networking Service Research Infrastructure Testbed]; by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2021R1I1A3060565); and by the MSIT (Ministry of Science and ICT), Korea, under the Innovative Human Resource Development for Local Intellectualization support program (IITP-2023-RS-2022-00156287) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

Data Availability Statement: Not applicable.

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

References

1. Nadir, Z.; Taleb, T.; Flinck, H.; Bouachir, O.; Bagaa, M. Immersive services over 5G and beyond mobile systems. *IEEE Netw.* **2021**, *35*, 299–306. [CrossRef]
2. Almasan, P.; Ferriol-Galmés, M.; Paillisse, J.; Suárez-Varela, J.; Perino, D.; López, D.; Perales, A.P.P.; Harvey, P.; Ciavaglia, L.; Wong, L.; et al. Digital twin network: Opportunities and challenges. *arXiv* **2022**, arXiv:2201.01144.
3. Wu, Y.; Zhang, K.; Zhang, Y. Digital twin networks: A survey. *IEEE Internet Things J.* **2021**, *8*, 13789–13804. [CrossRef]
4. Chen, H.; Dang, Z.; Hei, X.; Wang, K. Design and Application of Logical Range Framework Based on Digital Twin. *Appl. Sci.* **2023**, *13*, 6589. [CrossRef]

5. Raes, L.; Michiels, P.; Adolphi, T.; Tampere, C.; Dalianis, A.; McAleer, S.; Kogut, P. DUET: A Framework for Building Interoperable and Trusted Digital Twins of Smart Cities. *IEEE Internet Comput.* **2021**, *26*, 43–50. [CrossRef]
6. Ghaith, M.; Yosri, A.; El-Dakhakhni, W. Synchronization-Enhanced Deep Learning Early Flood Risk Predictions: The Core of Data-Driven City Digital Twins for Climate Resilience Planning. *Water* **2022**, *14*, 3619. [CrossRef]
7. Austin, M.; Delgoshaei, P.; Coelho, M.; Heidarinejad, M. Architecting smart city digital twins: Combined semantic model and machine learning approach. *J. Manag. Eng.* **2020**, *36*, 04020026. [CrossRef]
8. Liu, S.; Lu, Y.; Li, J.; Shen, X.; Sun, X.; Bao, J. A blockchain-based interactive approach between digital twin-based manufacturing systems. *Comput. Ind. Eng.* **2023**, *175*, 108827. [CrossRef]
9. Jwo, J.S.; Lee, C.H.; Lin, C.S. Data twin-driven cyber-physical factory for smart manufacturing. *Sensors* **2022**, *22*, 2821. [CrossRef]
10. Chen, X.; Liu, F.; Ai, Y.; Xu, G.; Chen, J.; Xu, X.; Liang, W. Key characteristics analysis of industrial digital twins for smart manufacturing. *Sci. Technol. Rev.* **2022**, *40*, 45–54.
11. Alnowaiser, K.K.; Ahmed, M.A. Digital Twin: Current Research Trends and Future Directions. *Arab. J. Sci. Eng.* **2022**, *48*, 1075–1095. [CrossRef]
12. Rusek, K.; Suárez-Varela, J.; Almasan, P.; Barlet-Ros, P.; Cabellos-Aparicio, A. Routenet: Leveraging graph neural networks for network modeling and optimization in sdn. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 2260–2270. [CrossRef]
13. Ferriol-Galmés, M.; Suárez-Varela, J.; Paillissé, J.; Shi, X.; Xiao, S.; Cheng, X.; Barlet-Ros, P.; Cabellos-Aparicio, A. Building a digital twin for network optimization using graph neural networks. *Comput. Netw.* **2022**, *217*, 109329. [CrossRef]
14. Azzouni, A.; Boutaba, R.; Pujolle, G. NeuRoute: Predictive dynamic routing for software-defined networks. In Proceedings of the 2017 13th International Conference on Network and Service Management (CNSM), Tokyo, Japan, 26–30 November 2017; pp. 1–6.
15. Zhou, C.; Yang, H.; Duan, X.; Lopez, D.; Pastor, A.; Wu, Q.; Boucadair, M.; Jacquenet, C. Digital Twin Network: Concepts and Reference Architecture (draft-irtf-nmrg-network-digital-twin-arch-03). In *IRTF Internet-Draft*; Internet Engineering Task Force: Fremont, CA, USA, 2023.
16. ITU-T. Digital twin network—Requirements and architecture (Y.3090). In *Telecommunication Standardization Sector of ITU*; ITU-T: Geneva, Switzerland, 2022.
17. Hui, L.; Wang, M.; Zhang, L.; Lu, L.; Cui, Y. Digital twin for networking: A data-driven performance modeling perspective. *IEEE Netw.* **2022**, *37*, 202–209. [CrossRef]
18. Wei, Z.; Wang, S.; Li, D.; Gui, F.; Hong, S. Data-driven routing: A typical application of digital twin network. In Proceedings of the 2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI), Beijing, China, 15 July–15 August 2021; IEEE: Piscataway, NJ, USA; pp. 1–4.
19. Isah, A.; Shin, H.; Aliyu, I.; Oh, S.; Lee, S.; Park, J.; Hahn, M.; Kim, J. A Data-Driven Digital Twin Network Architecture in the Industrial Internet of Things (IIoT) Applications. In Proceedings of the 10th International Conference on Advanced Engineering and ICT-Convergence, AEICP, Bangkok, Thailand, 7–10 February2023.
20. Gartner. *Rethink Network Monitoring for a Cloud Era*; Gartner: Singapore, 2018.
21. Yang, H.; Lü, P.; Sun, T.; Lu, L.; Zhou, C. Multi-source Heterogeneous Data Processing Technology for Digital Twin Network. In Proceedings of the 2022 IEEE 22nd International Conference on Communication Technology (ICCT), Nanjing, China, 11–14 November 2022; pp. 1829–1834.
22. Yan, Q.; Yu, F.R.; Gong, Q.; Li, J. Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 602–622. [CrossRef]
23. Kreutz, D.; Ramos, F.M.V.; Veríssimo, P.E.; Rothenberg, C.E.; Azodolmolky, S.; Uhlig, S. Software-Defined Networking: A Comprehensive Survey. *Proc. IEEE* **2015**, *103*, 14–76. [CrossRef]
24. Kim, H.; Feamster, N. Improving network management with software defined networking. *IEEE Commun. Mag.* **2013**, *51*, 114–119. [CrossRef]
25. Li, M.; Zhou, C.; Chen, D. Data Generation and Optimization for Digital Twin Network Performance Modeling (draft-li-nmrg-dtn-data-generation-optimization-00). In *IRTF*; Internet Engineering Task Force: Fremont, CA, USA, 2023.
26. Zhang, Q.; Ng, K.K.; Kazer, C.; Yan, S.; Sedoc, J.; Liu, V. MimicNet: Fast performance estimates for data center networks with machine learning. In Proceedings of the 2021 ACM SIGCOMM 2021 Conference, Online, 23–27 August 2021; pp. 287–304.
27. Ferriol-Galmés, M.; Paillisse, J.; Suárez-Varela, J.; Rusek, K.; Xiao, S.; Shi, X.; Cheng, X.; Barlet-Ros, P.; Cabellos-Aparicio, A. RouteNet-Fermi: Network Modeling With Graph Neural Networks. *IEEE ACM Trans. Netw.* **2023**. [CrossRef]
28. Yang, Q.; Peng, X.; Chen, L.; Liu, L.; Zhang, J.; Xu, H.; Li, B.; Zhang, G. Deepqueuenet: Towards scalable and generalized network performance estimation with packet-level visibility. In Proceedings of the ACM SIGCOMM 2022 Conference, Amsterdam, The Netherlands, 22–26 August 2022; pp. 441–457.
29. Shen, K.; Li, B. Learning-based Network Performance Estimators: The Next Frontier for Network Simulation. *IEEE Netw.* **2023**, 1.
30. Ge, Z.; Hou, J.; Nayak, A. Gnn-based end-to-end delay prediction in software defined networking. In Proceedings of the 2022 18th International Conference on Distributed Computing in Sensor Systems (DCOSS), Los Angeles, CA, USA, 30 May–1 June 2022; IEEE: Piscataway, NJ, USA; pp. 372–378.
31. Vinayakumar, R.; Soman, K.P.; Poornachandran, P. Applying deep learning approaches for network traffic prediction. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 13–16 September 2017; IEEE: Piscataway, NJ, USA; pp. 2353–2358.

32. Ramakrishnan, N.; Soni, T. Network traffic prediction using recurrent neural networks. In Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018; IEEE: Piscataway, NJ, USA; pp. 187–193.
33. Zebin, C.; Yichi, W.; Tang, H.; Chuanhuang, L. Research on intelligent perception model of sdn network delay. In Proceedings of the 2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS), Chengdu, China, 23–26 April 2021; pp. 452–457.
34. Fabien, G. Performance evaluation of network topologies using graph-based deep learning. *Perform. Eval.* **2019**, *130*, 1–16.
35. Yong, C.; Wei, Y.; Zhiyong, X.; Peng, L.; Zongpeng, D. Graph Neural Network Based Modeling for Digital Twin Network. IRTF, Internet-Draft. Available online: https://datatracker.ietf.org/doc/draft-wei-nmrg-gnn-based-dtn-modeling (accessed on 23 June 2023).
36. Siami-Namini, S.; Tavakoli, N.; Namin, A.S. The performance of LSTM and BiLSTM in forecasting time series. In Proceedings of the 2019 IEEE International conference on big data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; IEEE: Piscataway, NJ, USA; pp. 3285–3292.
37. Shohan, M.J.A.; Faruque, M.O.; Foo, S.Y. Forecasting of electric load using a hybrid LSTM-neural prophet model. *Energies* **2022**, *15*, 2158. [CrossRef]
38. Madan, P.; Singh, V.; Chaudhari, V.; Albagory, Y.; Dumka, A.; Singh, R.; Gehlot, A.; Rashid, M.; Alshamrani, S.S.; AlGhamdi, A.S. An optimization-based diabetes prediction model using CNN and Bi-directional LSTM in real-time environment. *Appl. Sci.* **2022**, *12*, 3989. [CrossRef]
39. Jovanovic, L.; Jovanovic, D.; Bacanin, N.; Jovancai Stakic, A.; Antonijevic, M.; Magd, H.; Thirumalaisamy, R.; Zivkovic, M. Multi-step crude oil price prediction based on lstm approach tuned by salp swarm algorithm with disputation operator. *Sustainability* **2022**, *14*, 14616. [CrossRef]
40. Lazaris, A.; Prasanna, V.K. Deep learning models for aggregated network traffic prediction. In Proceedings of the 2019 15th International Conference on Network and Service Management (CNSM), Halifax, NS, Canada, 21–25 October 2019; IEEE: Piscataway, NJ, USA; pp. 1–5.
41. Zhao, J.; Qu, H.; Zhao, J.; Jiang, D. Towards traffic matrix prediction with LSTM recurrent neural networks. *Electron. Lett.* **2018**, *54*, 566–568. [CrossRef]
42. Gao, Y.; Yin, D.; Zhao, X.; Wang, Y.; Huang, Y. Prediction of Telecommunication Network Fraud Crime Based on Regression-LSTM Model. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 3151563. [CrossRef]
43. Mekruksavanich, S.; Jitpattanakul, A. A multichannel cnn-lstm network for daily activity recognition using smartwatch sensor data. In Proceedings of the 2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering, Cha-am, Thailand, 3–6 March 2021; IEEE: Piscataway, NJ, USA; pp. 277–280.
44. Tao, S.; Cheng, Z.; Xiao-Dong, D.; Lu, L.; Dan-Yang, C.; Hong-Wei, Y.; Yan-Hong, Z.; Chao, L.; Qin, L.; Xiao, W.; et al. Digital twin network (DTN): Concepts, architecture, and key technologies. *Acta Autom. Sin.* **2021**, *47*, 569–582.
45. Zhu, Y.; Chen, D.; Zhou, C.; Lu, L.; Duan, X. A knowledge graph based construction method for Digital Twin Network. In Proceedings of the 2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI), Beijing, China, 15 July–15 August 2021; pp. 362–365.
46. Chen, D.; Yang, H.; Zhou, C.; Lu, L.; Lü, P.; Sun, T. Classification, Building and Orchestration Management of Digital Twin Network Models. In Proceedings of the 2022 IEEE 22nd International Conference on Communication Technology (ICCT), Nanjing, China, 11–14 November 2022; pp. 1843–1846.
47. China Mobile Research Institute. *Digital Twin Network(DTN) White Paper*; China Mobile Research Institute: Beijing, China, 2021.
48. Zhao, J.; Xiong, X.; Chen, Y. Design and Application of a Network Planning System Based on Digital Twin Network. *IEEE J. Radio Freq. Identif.* **2022**, *6*, 900–904. [CrossRef]
49. Kim, J.; Kang, C. Anomaly detection of railway vehicle screw air compressors through data feature extraction. *J. Korean Soc. Mech. Eng. Ser. A* **2023**, *47*, 489–496.
50. López Brescó, A.; Suárez-Varela, J.; Ferriol-Galmés, M.; Cabellos-Aparicio, A.; Barlet-Ros, P. Network Modeling Datasets. Available online: https://github.com/BNN-UPC/NetworkModelingDatasets/tree/master (accessed on 31 July 2023).