*electronics*

*Article*

# Machine Learning with Adaptive Time Stepping for Dynamic Traffic Load Prediction in 6G Satellite Networks

**Yangan Zhang [1,*], Xiaoyu Zhang [1], Peng Yu [2] and Xueguang Yuan [1]**

1 State Key Laboratory of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications, Beijing 100876, China; zhang_xy@bupt.edu.cn (X.Z.); yuanxg@bupt.edu.cn (X.Y.)
2 State Key Lab Networking & Switching Technol, Beijing University of Posts and Telecommunications, Beijing 100876, China; yupeng@bupt.edu.cn
* Correspondence: zhang@bupt.edu.cn

**Abstract:** The rapid development of sixth-generation (6G) mobile broadband networks and Internet of Things (IoT) applications has led to significant increases in data transmission and processing, resulting in severe traffic congestion. To better allocate network resources, predicting network traffic has become crucial. However, satellite networks face global imbalances in IoT traffic demand, with substantial variations in satellite density and load distribution within the same constellation. These disparities render traditional traffic prediction algorithms inadequate for dynamically changing satellite network topologies. This paper thoroughly examines the impact of adaptive time stepping on the prediction of dynamic traffic load. Particularly, we propose a high-speed traffic prediction method that employs machine learning and recurrent neural networks over the 6G Space Air Ground Integration Network (SAGIN) structure. In our proposed method, we first investigate a variable step size-normalized least mean square (VSS-NLMS) adaptive prediction method for transforming time series prediction datasets. Then, we propose an adaptive time stepping-Gated Recurrent Unit (ATS-GRU) algorithm for real-time network traffic prediction. Finally, we compare the prediction accuracy of the ATS-GRU algorithm with that of the fixed time stepping-Gated Recurrent Unit (FTS-GRU) algorithm and compared the prediction results of three different step sizes (FSS, VSS, and ATS) based on normalized least mean square (NLMS). Numerical results demonstrate that our proposed scheme can automatically choose a suitable time stepping to track and predict the traffic load curve with acceptable accuracy and reasonable computational complexity, as its time stepping dynamically adjusts with the traffic.

**Keywords:** satellite network; dynamic traffic load prediction; adaptive time stepping (ATS); Gate Recurrent Unit (GRU); machine learning

check for updates

**Citation:** Zhang, Y.; Zhang, X.; Yu, P.; Yuan, X. Machine Learning with Adaptive Time Stepping for Dynamic Traffic Load Prediction in 6G Satellite Networks. *Electronics* **2023**, *12*, 4473. https://doi.org/10.3390/electronics12214473

Academic Editor: Martin Reisslein

Received: 12 September 2023
Revised: 19 October 2023
Accepted: 25 October 2023
Published: 31 October 2023



## 1. Introduction

As the global mobile industry progresses towards the era of 6G networks, satellite networks are increasingly emerging as a primary competitive technology for delivering ubiquitous communication services worldwide. These satellite networks serve as a valuable complement to terrestrial infrastructure, offering an extensive coverage area and ensuring system fault tolerance. This resilience enables users to maintain connectivity in remote regions or during natural disasters when terrestrial communication may be unavailable. Additionally, satellite networks facilitate the integration of cutting-edge technologies such as Artificial Intelligence, IoT, Big Data, and others, enabling the provision of a diverse range of services. When compared to Wi-Fi, Bluetooth, and traditional cellular networks, satellite networks possess distinct advantages for remote facility monitoring and real-time asset management across vast geographical expanses. Currently, satellite communication networks have found application in global communication, high-precision navigation and positioning (e.g., Bei Dou Navigation Satellite System), remote sensing, and various other scenarios. These applications serve as compelling evidence of the effectiveness and full capability of utilizing satellite networks for communication and data transmission interactions.

In satellite networks, an ingress satellite gathers data from terminals and forwards it to destination terminals through egress satellites. Satellite networks offer a convenient solution for terminal devices located in remote areas lacking stable terrestrial broadband infrastructure, granting them easy access to the cloud. Likewise, cloud operators can effortlessly distribute new configurations and tasks to distributed terminal devices, regardless of location or time [1]. Efficient data transmission in satellite networks relies heavily on an effective routing scheme. Nevertheless, on a global scale, traffic requirements within satellite networks are often unevenly distributed. Satellites covering developed regions generally handle heavier traffic loads compared to those covering underdeveloped areas, oceans, or mountainous regions [2]. Furthermore, when traditional shortest path routing algorithms are employed, irregular constellation structures and varying inter-satellite link (ISL) lengths can lead to traffic congestion in specific areas. Furthermore, the issues related to communication performance, energy efficiency, and security in integrated satellite and ground terminal networks have also sparked discussions among scholars [3–5]. Given the limited bandwidth resources typically available from satellites, these factors can quickly result in severe congestion and significant packet drops in certain satellites while leaving others underutilized. Hence, the implementation of rational and effective resource allocation mechanisms, taking into account satellite bandwidth resources and dynamic traffic demands, becomes crucial for meeting user service requirements. Resource allocation directly impacts the operational costs of satellite networks and the user experience of IoT devices. An irrational allocation of resources not only increases communication network operating costs but can also lead to substantial energy wastage. Accurately predicting network traffic trends offers an insightful representation of shifting service demands. The precision of prediction outcomes is paramount for achieving a well-balanced allocation of network resources, optimizing the distribution of existing satellite resources, and efficiently offloading computing tasks [6,7]. Consequently, establishing a high-precision traffic prediction model for satellite networks holds significant practical importance.

Analyzing the distribution and forecasting the demand for communication traffic to guide the allocation of communication resources has consistently been a significant research area [1]. Traditional approaches for wireless traffic prediction have typically relied on manual and statistical methods. However, these conventional traffic prediction methods are no longer suitable for satellite networks like SAGIN due to their dynamic changes in topology and link status. Recently, emerging machine learning (ML) technologies have displayed remarkable advantages in optimizing dynamic routing. ML enables the real-time perception of dynamic network traffic conditions, facilitating on-the-fly adjustments to network routing policies.

Looking ahead, artificial intelligence is poised to become an intrinsic feature of 6G networks. AI-based traffic prediction models can autonomously extract various features from wireless data and use them comprehensively for precise real-time wireless traffic predictions. In the context of 6G satellite networks, this AI integration enables enhanced network autonomy and intelligence, leading to intelligent operation and maintenance management of terminal devices, network routing policies, and efficient automated service deployments [8,9]. The evolution toward an intelligent network underscores the importance of traffic analysis and precise demand forecasting. Anticipating user demands in advance enables the network to allocate resources more efficiently, ensuring swift resource distribution among users contending for these resources.

The remainder of this paper is structured as follows. Section 2 provides an overview of related works closely aligned with our research objectives. In Section 3, we present a business load prediction model for 6G satellite networks, taking SAGIN as an illustrative example. Section 4 delves into traffic load prediction methodologies employing adaptive time stepping (ATS), incorporating machine learning and recurrent neural network (RNN) techniques. Within this section, we introduce the ATS-NLMS adaptive prediction method, which serves to transform time series prediction datasets, and the ATS-GRU algorithm for real-time network traffic prediction. Section 5 is dedicated to the presentation of simulations and an analysis of the obtained results. Finally, Section 6 concludes the full text.

## 2. Related Works

In recent years, research into traffic allocation methods has undergone a shift from static allocation approaches to predictive dynamic allocation methods. Effectively and rapidly predicting future network traffic, whether in the short term or long term, has emerged as a critical aspect of traffic planning in satellite communication networks [10]. Conventional models like autoregressive (AR), moving average (MA), autoregressive moving average (ARMA), and autoregressive integrated moving average (ARIMA) models are limited in their ability to capture only short-term traffic correlations. With the continuous integration of technologies such as neural networks and support vector machines, prediction models based on machine learning algorithms have surfaced. These encompass artificial neural networks, least squares support vector machines (LSSVM), and extreme learning machines (ELM), among others [11]. However, these algorithms have faced challenges related to the absence of temporal correlation consideration in time series data, limited prediction accuracy, and ineffective prediction of satellite network traffic [12]. Given the intensive research into machine learning techniques, recent efforts have explored the application of machine learning and deep learning algorithms for predicting satellite network traffic.

Li N, Hong Z J, L. Yang, etc., in [13–17] collectively addressed the challenge of network traffic prediction within satellite networks. Among these studies, with the exception of [14], the adoption of Gated Recurrent Unit (GRU) models has demonstrated superior prediction performance compared to other algorithms such as ARIMA, FARIMA, Support Vector Regression (SVR), Random Forest, and traditional recurrent neural networks (RNNs), among others. To address the issue of limited online traffic data availability, ref. [13] employs a transfer learning approach in conjunction with the GRU neural network traffic prediction algorithm, resulting in precise predictions of satellite network traffic. Additionally, ref. [15] introduces an attention mechanism to enhance the autocorrelation by combining it with the GRU neural network, thereby enhancing prediction accuracy. The Long Short-Term Memory (LSTM) model in [17] balances the influence of different input components on the output by incorporating an attention mechanism. Furthermore, in order to incorporate spatial characteristics into the analysis, ref. [16] combines GRU with Graph Convolutional Networks (GCNs), utilizing GCN for preprocessing to extract link features for subsequent training.

In [18], the author presents a kernel least mean square algorithm (KLMS) with adaptive step length and adaptive kernel width, which maps the nonlinear data from low-dimensional input space to high-dimensional feature space, and the algorithm will adaptively adjust the step length and kernel width based on the instantaneous error in the iterative process, which will provide strong decision support for traffic planning and routing design in the satellite network. Additionally, in [19], wavelet transform is employed as a preprocessing step, transforming one-dimensional time series data into three-dimensional data, which facilitates feature extraction for GRU and ultimately yields superior performance compared to RNNs.

As indicated in the preceding discussion, it becomes evident that GRU outperforms other algorithms when it comes to forecasting time series data, including network traffic. However, conventional approaches to traffic prediction typically treat data sampling and prediction as separate entities. Consequently, the sampling process, which frequently employs a fixed time interval, tends to overlook variations in traffic speed. This can result in computationally intensive procedures and issues related to prediction accuracy, making such methods unsuitable for deployment in the context of advanced 6G satellite networks exemplified by ASIGN. To address these aforementioned challenges, this paper introduces a novel algorithm named ATS-GRU, which effectively combines adaptive time stepping with GRU, providing a solution that enhances prediction accuracy and computational efficiency.

## 3. Architecture Design

### 3.1. Satellite Routing

Recently, MLSNs (Multilayered Satellite Networks) have been put forward as a feasible design for next-generation satellite networks. Hierarchical architecture and many satellite networks are combined to create MLSNs. A typical bilayer MLSN consists of a low earth orbit (LEO) constellation and a medium earth orbit (MEO) constellation interconnected by intersatellite connections (ISLs) to create a mesh or ring topology. Furthermore, interlayer linkages (ILLs) connect satellites at various levels. Ground-satellite linkages (GSLs) allow terrestrial users to connect satellites at various levels. By integrating these diverse networks, we can improve the capacity, dependability, and performance of global communications. Figure 1 shows a portion of a multilayer satellite network, where GEO, MEO, and LEO satellites cover the terrestrial gateways. Packets are routed independently to the target gateway after being delivered to the satellite network across a number of satellite hops, perhaps in various layers. Routing decisions made in satellite networks are separate from those in the terrestrial networks.

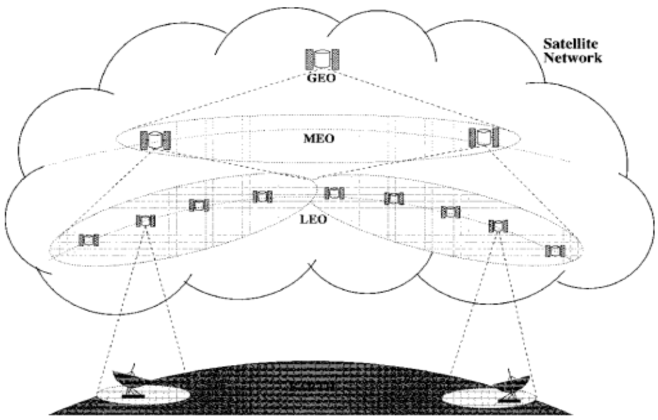


**Figure 1.** Multilayered satellite network.

The satellite network is hierarchically structured to decrease computational complexity and reduce communication burden across the network. Satellites are organized into groups and managed by a satellite in the highest tier in this structure. Routing table computations are performed using a hierarchical structure.

### 3.2. Traffic Load Prediction

The satellite communication network connects the space and terrestrial networks by organizing satellites as network nodes. The advancements in on-satellite processing techniques and inter-satellite communication technologies have greatly increased the ability of satellites to analyze information autonomously and communicate directly with each other. To enable information sharing across coverage areas, satellite network routing strategies are responsible for transmitting and distributing inter-satellite data and providing the required Quality of Service (QoS) for various services. As a result, research into satellite network routing strategies is crucial for further advancements in the field.

Routing algorithms for satellite networks have two main aspects: computing paths with low communication and computation overhead, and adjusting routing choices in real time with network topology changes. A good routing strategy is essential for transmitting large volumes of data across satellite networks. Globally, the demand for IoT traffic from satellite networks is imbalanced, with more traffic in industrialized regions than in less developed areas, mountains, or seas [20]. However, using the conventional shortest path routing algorithm may lead to traffic build-up in some regions due to variable inter-satellite link (ISL) lengths and asymmetric constellation topologies, such as the lowest horizontal ring [18]. Load-balancing routing techniques have been proposed to address this issue. These techniques typically employ either a local strategy, where satellites independently choose their routes based on local traffic data, or a global strategy, which uses data on the condition of all traffic globally to make routing decisions from a global perspective. While neither strategy provides the best performance, they offer different advantages and disadvantages [21].

Three main methods of performing load balancing based on routing already exist according to existing technology: source-based, centralized, and distributed load balancing systems [22]. The distributed load balancing approach operates on the idea that each satellite independently chooses the best way to transmit data, allowing them to react quickly to fluctuations in traffic.

Load-balancing routing techniques have another fundamental flaw in that they proactively distribute traffic to avoid congestion while only passively doing load balancing to alleviate detected congestion. The fact that these systems are created under the assumption that traffic demand is unknown and random may be an underlying reason for this behavior.

The majority of satellite routing algorithms currently available are based on those designed for terrestrial networks. Typically, they rely on either the shortest path or the concept of minimum cost.

Therefore, it is necessary to study and develop dynamic routing to solve the difficulties encountered in the development of satellite networks. The variation in satellite density across latitudes and differences in user distribution result in significant load imbalances within satellite constellations. Further complicating matters, high-load coverage areas shift rapidly due to the high speed of satellite movement. As a result, traditional routing algorithms have become increasingly insufficient for the demands of modern satellite networks.

Demands for information transmission and collection have outstripped the capabilities of ground-based communication networks. As a result, research on satellite networks has gained popularity. This is due to their ability to overcome geographical limitations and provide an inexpensive, stable, and reliable means of communication.

The resolution of inter-satellite routing problems is a prerequisite for utilizing satellite networks. The performance of routing has a direct impact on all aspects of satellite network operation, making it imperative to find an effective solution.

### 3.3. Overview of Architecture Design

Space Air Ground Integration Network (SAGIN), due to its advantages in global coverage, high dynamism, energy efficiency, and security, has become one of the crucial application scenarios for traffic load prediction research in 6G satellite networks. As shown in Figure 2, SAGIN's traffic prediction system model based on machine learning incorporates air networks, satellite systems, and ground networks. The space layer consists of communication satellites distributed across different orbits. The aerial layer consists of diverse communication devices, including airships and unmanned aerial vehicles (UAVs), while satellite networks encompass satellites with varying altitudes, classified as GEO, MEO, and LEO. The ground layer is composed of multiple ground gateways, various ground communication networks, and numerous end-users with different end devices. Lower satellites in the network may be obstructed by higher satellites. Satellite networks are classified as a separate AS (Automatic System). As mentioned in the first and second sections, different researchers have mostly used scenarios involving low earth orbit satellite networks, two-tier satellite networks, or other 5G mobile networks for traffic prediction. These networks typically operate with stationary satellite orbits and predefined coverage areas, constraining network flexibility and leading to increased communication latency and diminished real-time performance. However, SAGIN networks are inherently more complex and dynamic. Multi-tiered components interact in a continuously evolving environment. This heightened complexity and ever-changing conditions demand advanced traffic prediction techniques to effectively manage network resources, a challenge not typically encountered in conventional satellite networks. Therefore, to achieve load balancing across the 6G network system, we present a traffic load prediction model using machine learning and the SAGIN structure in this study.

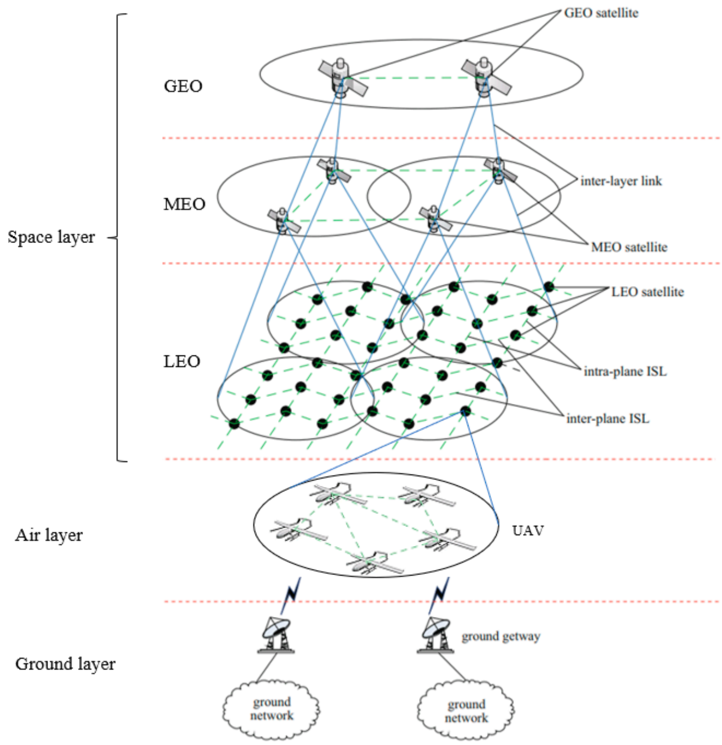


**Figure 2.** Structure of traffic prediction system model.

As different terminal types and usage scenarios lead to varying demands on computing, storage, and communication capabilities, the capacity wealth of some network nodes can be severely limited or wasted. Therefore, to achieve the aforementioned goals, the system must allocate network bandwidth resources appropriately, irrespective of the terminal type or usage scenario.

Although compared with 5G, the propagation speed and various indicators of 6G have improved significantly, but the lack of network resources is an eternal problem; in particular, the competition for bandwidth resources still exists and has an increasingly fierce trend. Accurately predicting traffic patterns can optimize resource allocation and balance network load, leading to enhanced network performance. Dynamic control of resources according to different demands of different user terminals is the premise and guarantee of realizing dynamic resource allocation. In order to further utilize bandwidth resources, we must improve real-time control of network resources according to communication needs and thus improve utilization, a goal in which real-time estimation of traffic plays an important role.

## 4. Model Design

Traditional traffic prediction methods such as ARIMA, SVM, ELM, RNN, etc., usually separate the sampling and prediction processes, treating them as two distinct, independent procedures. Unfortunately, the use of constant time stepping of sampling in these methods can often lead to excessive sampling and decreased efficiency. Although the literature does not explicitly state it, using a fixed sampling rate in traffic prediction methods that do not take into account the changes in traffic speed may lead to computational complexity and inaccuracies in traffic predictions. To optimize resources and reduce costs, our innovative solutions must account for computational efficiency and prediction performance while also being tied to cost and public good estimations.

This article showcases the implementation of adaptive time steps in a GRU prediction model and introduces a simple and easy method for determining time step divisors for highly dynamic traffic load events. Additionally, this paper proposes a new algorithm, referred to as ATS-GRU, that merges adaptive time stepping and GRU. Our method employs a feedback loop where the prediction algorithm error is used to control the time stepping, considerably decreasing overhead while still ensuring traffic prediction accuracy.

### 4.1. Mathematical Description

Predicting network traffic load within the context of a Space Air Ground Integration Network (SAGIN) is an intricate and multifaceted challenge. This process entails forecasting future network traffic load by drawing from historical data and pertinent parameters that are unique to such network environments. It can be envisioned as the task of predicting network traffic patterns within a specified geographical area and time horizon, with the primary objective of resource allocation optimization and the promotion of efficient communication. Below, we provide a mathematical delineation of this problem.

Let $T$ represent time, where $t_1, t_2, \cdots, t_n$ are discrete time points. Let $L(t)$ represent the traffic load at time $t$ in the SAGIN. This load could be measured in bits per second (bps), packets per second (pps), or other relevant units. Let $X(t)$ represent a vector of predictor variables at time $t$, which could include factors such as satellite bandwidth, air-to-ground link conditions, ground station congestion, and the number of active users.

Given historical data of traffic load and associated predictor variables, the problem is to find a function $f$ that predicts the traffic load $L(t)$ at a future time $t + \Delta t$ based on the predictor variables $X(t + \Delta t)$. Similar to the general network traffic prediction problem, this can be framed as a regression problem. A common approach is linear regression, where $f$ can be represented as

$$L(t + \Delta t) = \beta_0 + \beta_1 \cdot X_1(t + \Delta t) + \beta_2 \cdot X_2(t + \Delta t) + \cdots + \beta_p \cdot X_p(t + \Delta t) + \varepsilon \tag{1}$$

$\beta_i$ are the coefficients to be estimated. $X_i(t + \Delta t)$ are the predictor variables at the future time $t + \Delta t$. $\varepsilon$ is the error term.

Historical data specific to the SAGIN environment are used to train the model. Coefficients $\beta_i$ are estimated to minimize the prediction error, often by minimizing the mean squared error (MSE) or another suitable loss function. The model's performance is assessed using metrics like Root Mean Squared Error (RMSE) on a validation dataset. Once trained and evaluated, the model can be deployed to predict future traffic loads based on the predictor variables. Traffic load prediction in SAGIN, as in other network contexts, may require periodic model updates as new data become available. This iterative process ensures that the predictions remain accurate in a dynamic network environment.

The specifics of the mathematical model and predictor variables should be tailored to the unique characteristics and objectives of a Satellite–Air–Ground Network. Factors like satellite positioning, link quality, mobility patterns, and ground station capacities are crucial variables to consider when building and training the prediction model for SAGIN.

### 4.2. Adaptive Time-Stepping Forecasting Method

In choosing a suitable time step for a model, it is important to not only ensure numerical stability but also to ensure that the time step is sufficiently small to mitigate the risk of missing the peak of a wave at a sampling point. The chosen computation interval provides valuable opportunities to characterize the traffic load, but the value of these opportunities varies over the course of the simulation and should be leveraged accordingly. To further illustrate this point, consider the example of converting analog music to digital, as shown in Figure 3. The wave must be sampled at regular intervals, indicated by the vertical green arrows. A greater number of arrows per sampling point increases the number of interpolation points, resulting in a more defined wave, while fewer arrows provide less definition.

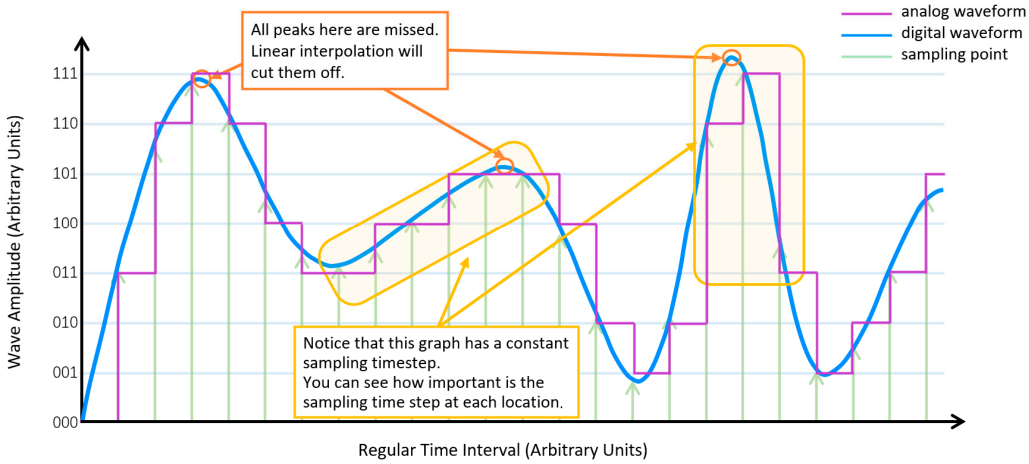


**Figure 3.** Relationship between computational time step and wave definition.

Is it possible to have too many arrows when sampling a wave? While more arrows lead to higher quality, a point is reached where the increased quality is not discernible by the human ear. In Figure 3, selected areas are highlighted to demonstrate the potential for adaptive time stepping. Adaptive time stepping optimizes arrow placement using a waveform objective function. This can be compared to running a GRU model at a uniform 30 s step, versus advanced time step control that automatically adjusts using the ATS algorithm. The adaptive time stepping reduces sampling without sacrificing accuracy by identifying areas where more information is needed and sets step sizes accordingly.

It is important to note that regular interval sampling is not required to accurately reproduce the wave later. Rather, identifying instances when greater definition is necessary is key in determining an appropriate sampling strategy. To minimize sampling overhead, one simple adaptive rate strategy is to use a low sampling rate during slow periods and a high rate during fast periods of change. The goal of this adaptive interval is to efficiently sample and reconstruct the traffic curve without compromising prediction accuracy, as would be the case with a constant rate approach.

In the following section, we examine a typical IoT traffic curve scenario. The curve is divided into slow- and fast-changing periods based on $T^s_{\max}$ and $T^f_{\max}$, representing the maximum step size. As per Nyquist sampling theory, the slow-changing zone mandates a step size less than $2T^s_{\max}$, while the fast-changing zone demands a step size under $2T^f_{\max}$. Subsequently, an average step size of $S^{avg}_{ATS}$ can then be achieved by

$$S^{avg}_{ATS} = \frac{T_s + T_f}{2T_s/T^s_{\max} + 2T_f/T^f_{\max}} \tag{2}$$

where $T_s$ and $T_f$ denote the duration of slow-changing and fast-changing periods, respectively. Since $T^s_{\max} > T^f_{\max} > T_{\max}$, we deduce $S^{avg}_{ATS} > 2T_{\max}$, which indicates that the ATS approach results in significantly reduced sampling overhead compared to its constant-step-size counterpart.

### 4.3. Linear Predictor with Adaptive Time Stepping

Choosing the appropriate time-stepping size when utilizing a fixed-size LMS linear predictor for traffic load prediction can prove challenging. A larger step size may result in quicker convergence and shorter response time to traffic load changes, but such an approach could result in substantial weight fluctuations in the predictor following convergence. On the contrary, a smaller step size yields slower convergence but a more stable predictor in terms of weight fluctuations. However, due to the dynamic of traffic load changes in 6G satellite networks, fixed-step-size methods are often unable to deliver favorable predictive performance consistently. An Adaptive Time stepping (ATS) LMS linear predictor is, therefore, a more suitable approach for such applications.

The ATS-LMS adaptive predictor adjusts its step size in real time, based on the current squared prediction error. Typically, a larger squared error value results in an increase in the step size, providing faster tracking, while a smaller squared error value leads to a decrease in the step size, resulting a in smaller misadjustment.

### 4.4. RNN and GRU

Traffic control relies on precise and real-time flow prediction, and researchers have started applying deep NN methods such RNN to traffic predictions [23]. By employing RNN, which was first used for language models due to its ability to preserve long-term dependencies, researchers found that the gradients in RNN can vanish when dealing with longer time lags. To address this issue, LSTM and GRU, two types of RNN structures with forget units, were developed to enable memory cells to determine the information to discard while preserving the optimal time lags. LSTM, a model best known for its ability to remember long-term dependencies, was initially created in 1997 as a language model [24]. Although it is complex, LSTM is regarded as time-consuming to train. In 2014, a minor modification to LSTM was proposed, known as GRU, which enhances machine translation performance [25]. The Gate Recurrent Unit (GRU) is a type of recurrent neural network (RNN) that reduces matrix multiplication time without compromising performance on small datasets. Its structure is similar to that of the Long Short-Term Memory (LSTM) network; however, in GRU, the forget gate and input gate are connected to form a single update gate [26]. Figure 4 depicts a fundamental GRU unit design.

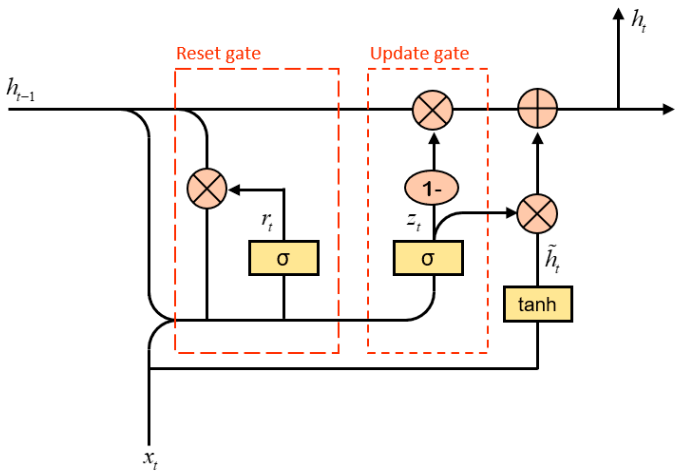


**Figure 4.** The internal structure of the GRU model.

The GRU formulas are as follows:

$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right) \tag{3}$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right) \tag{4}$$

$$\widetilde{h}_t = \tanh\left(W_{\widetilde{h}} \cdot [r_t * h_{t-1}, x_t]\right) \tag{5}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h}_t \tag{6}$$

where $x_t$ is the input vector of training data at time $t$, and $h_t$ is the previous layer output at time $t$. $z_t$ and $r_t$ are the reset and update gates, respectively. The candidate activation is represented by $\widetilde{h}_t$.

### 4.5. The Proposed ATS-GRU Model

Let $B_{n-1:n}$ be the average traffic loads from a satellite network at time slot $\Delta t_{n-1}$, $n = t_n - t_{n-1}$. The conventional approach conducts traffic sampling at a constant rate, as follows:

$$\Delta t_{0,1} = \Delta t_{1,2} = \cdots = \Delta t_{n-1,n} = \cdots , \ (n = 0, 1, 2, \cdots)$$

The management layer of the satellite network, upon accurately receiving the load value of $B_{n-1:n}$, can dynamically adjust routing strategies to better allocate network resources and manage service quality. At the next time slot $\Delta t_n$, satellite traffic load may change to a new value of $B_{n:n+1}$, and the management layer will have to make adjustments to the routing strategy once again.

It is, nevertheless, impossible for the manager to receive the value of $B_{n-1:n}$ before the time of $t_n$. Thus, at time $t_{n-1}$, we can estimate the approximate value $B_{n-1:n}$, which is denoted by $B'_{n-1:n}$. If $B'_{n-1:n} < B_{n-1:n}$, the satellite link or node may reach its capacity limit and suffer from a shortage of bandwidth resource to accommodate all streaming sessions. As a solution, it is possible to optimize routing strategies to ensure that traffic is transmitted via the shortest or optimal paths or to evenly distribute traffic across multiple satellite nodes.

On the other hand, if $B'_{n-1:n} > B_{n-1:n}$, part of the bandwidth resource is simply overbooked and useless. In this sense, the traffic prediction is critical to ensure a good performance of our proposed scheme.

The 6G satellite network traffic load prediction model requires two key components:

(1) a way to accurately calculate and predict network traffic load;
(2) a method to determine and predict the appropriate length of time stepping at any given time, whether long or short step sizes are required.

For a 6G satellite network, the traffic load varies in such a dynamic way that a fixed step size can hardly achieve a good performance all the time. For this reason, the adaptive time-stepping method combined with GRU neural networks is more suitable for predicting the dynamic traffic load of a satellite network. The ATS-GRU predictor adjusts the step size dynamically according to the square of prediction errors. Generally speaking, a greater squared value will cause ATS-GRU to increase the time step size to provide faster tracking, while a smaller squared value will cause ATS-GRU to decrease the time step size to obtain a smaller misadjustment.

The GRU model takes as input a time window of traffic load data, represented as $X(n) = [Y_{n-m}, Y_{n-m+1}, \cdots, Y_n]$, with $m$ representing the size of the time window. These data are processed by the GRU model to generate a traffic load prediction for the next time $t_{n+1}$ point, denoted as $\hat{Y}_{n+1}$. Subsequently, we can assess the prediction accuracy using various evaluation metrics, resulting in the prediction error $e(n)$. ATS-GRU rules the time-stepping size at time $t_n$ as $T_n$, and then updates it in integration by

$$S'_s(n+1) = S_s(n)\left[\gamma + (1-\gamma)\frac{|e(n)|}{E_b}\right] \tag{7}$$

with $0 < \gamma < 1, \ E_b > 0$, and

$$S_s(n+1) = \begin{cases} S_s^{\max}, & \text{if } S'_s(n+1) > S_s^{\max} \\ S_s^{\min}, & \text{if } S'_s(n+1) < S_s^{\min} \\ S'_s(n+1), & \text{otherwise} \end{cases} \tag{8}$$

where $\gamma$ is the remembering factor and is usually chosen in the range (0, 1) to provide exponential forgetting, $E_b$ is the targeted prediction error bound, and $S_s^{\max}$ and $S_s^{\min}$ are the upper and lower bounds for time-stepping size. We will give a more detailed explanation of the parameters next.

The scheme of ATS-GRU optimizes the sampling interval by the following principle. The larger the prediction error is, the sharper the traffic load curve changes. As a result, we must increase the time-stepping size for satisfying prediction accuracy. In contrast, the smaller the prediction error is, the smaller the time-stepping size becomes. In particular, the value of the targeted prediction error bound $E_b$ can be defined and adjusted by the manager based on different network requirements or application needs. If $|e(n)| > E_b$, the time-stepping size should be increased; if $|e(n)| < E_b$, the time-stepping size should be decreased; if $|e(n)| = E_b$, the time-stepping size should be kept unchanged.

$S_s^{\max}$ is the upper bound of the time-stepping size, which represents a low sampling rate, leading to the bottom-line responding time of the predictor. $S_s^{\min}$ is the lower bound of the time-stepping size, which can lead to lower prediction error but larger computational complexity as the value rises. $\gamma$ decides how much the current time-stepping size is relying on its previous value rather than the prediction error. We have to leverage the value of remembering factor $\gamma$ for the optimal performance of the predictor, with $\gamma = 70\%$ as a typical case.

In deciding $S_s^{\max}$, we are faced with the choice between prediction accuracy and compilation efficiency. High values of $S_s^{\min}$ lead to decreased prediction errors, but also an increased computational complexity of the prediction method. Selection of $S_s^{\max}$ implies that the linear predictor must react quickly enough to manage burst traffic.

In SAGIN, each satellite layer serves distinct purposes, providing global communication, positioning, and data transmission services. It supports various applications such as the internet, weather monitoring, military communication, and emergency response. The system is influenced by dynamic factors like weather conditions in different geographical regions, fluctuations in user numbers and locations, varying application demands, and potential communication interferences like solar radiation and changes in the earth's atmospheric conditions. In this complex network environment, accurately predicting dynamic traffic loads helps optimize resource management, including rational allocation of satellite bandwidth and processing capabilities, to reduce resource wastage. These predictions can be used to adjust bandwidth allocation to compensate for adverse weather effects on signal transmission, ensuring uninterrupted service. Managing changes in user numbers and locations helps avoid network congestion and unstable signals, prompting proactive measures such as increasing bandwidth or adjusting satellite routes. Additionally, precise traffic load predictions aid in cost savings, reducing energy consumption and enhancing the economic efficiency of network operations. Overall, this forecasting contributes to improving the efficiency of the integrated satellite network in air and space and on earth.

Our proposed ATS-GRU algorithm is divided into three primary components: data transformation and partitioning into training and test sets, training of the GRU network, and traffic flow prediction and verification. The model's critical steps are enumerated as follows:

Step 1: The traffic flow data that were collected are partitioned into training and test sets, with a 7:3 ratio.

Step 2: The one-dimensional traffic data are transformed into two-dimensional data based on the "sliding window" principle.

Step 3: The raw data are pre-processed by utilizing multiple techniques, including normalization.

Step 4: The training data are fed into the GRU network, operationalizing the optimization of the model parameters used to construct the prediction model.

Step 5: The trained GRU neural network is evaluated using test input vectors by generating predicted results in comparison to the actual values, where the resulting model error is subsequently calculated.

## 5. Numerical Results

This section presents the implementation of the proposed ATS-GRU algorithm, followed by its comparison with the traditional fixed time-stepping algorithm, FTS-NLMS. To ensure a just comparison of performance, we implement the original fixed time-stepping algorithm, FTS-NLMS, with the GRU technique, referred to as FTS-GRU. In addition, to further explain and illustrate the importance and advantages of the ATS method in traffic load prediction, we also compared the prediction results of three different step sizes (FSS, VSS, and ATS) based on NLMS.

### *5.1. Evaluation Setup*

We assess the efficiency of the proposed architecture by employing mobile traffic statistics collected from ten discrete Mobile Edge Computing (MEC) servers spanning one month. There are multiple factors that can impact the precision of predictions, including the validity of data sources, techniques used for prediction, and experimental conditions. Therefore, evaluating the forecasting model mandates the appropriate selection of indicators. Accuracy functions are the essential key performance indicator (KPI) that measures the forecasting model's quality directly. The study employs Normalized Root Mean Square Error (NRMSE) to evaluate the forecasting model, which is defined as

$$NRMSE = \frac{1}{y}\sqrt{\frac{1}{N}\sum_{i=1}^{n}(y_i - \widetilde{y_i})^2} \tag{9}$$

where $y_i$ denotes the actual value, while $\widetilde{y_i}$ symbolizes the predicted value of $y_i$, with $N$ representing the total number of testing samples.

Python served as the programming language of choice for conducting the simulation, with the TensorFlow and Keras backends employed. Table 1 features the hyperparameters selected for this study. One of the essential hyperparameters that must be defined is the number of hidden layers, which can influence the trade-off between time and accuracy required for training the network. For this study, we have fixed the number of hidden layers to 5. The amount of data used to train the network is a crucial determinant of the prediction accuracy. Adding more layers may help to enhance the prediction accuracy but may also increase the training time. To ensure a reasonable balance between training time and accuracy, the total number of epochs was set to 100. The model undergoes training and validation using three weeks of data, and the outcomes summarized for the last week. Adam optimization guides the iterative update of network weights using training data.

**Table 1.** Training hyperparameters.

| **Hyperparameters Name** | **Value** |
|---|---|
| Initial Learning Step Size | 100 |
| Number of Epochs | 100 |
| GRU Hidden States | 64 |
| GRU Hidden Layers | 5 |
| Optimization Algorithm | Adam |
| Loss Function | MAE |

### *5.2. Analysis of Results*

The paper demonstrates the results of multi-step prediction, where predictions for future time intervals are inferred by postponing the output by a predetermined number of time slots. It is demonstrated that the accuracy of predicting traffic statistics for upcoming time steps decreases. Additionally, the paper evaluates the influence of timeslot duration T and the number of GRU network observations on prediction outcomes. To achieve accurate predictions in traffic data analysis, it is essential to determine the GRU network's memory storage capacity and minimum traffic data requirements. These parameters impact the network's ability to store information and provide reliable predictions.

We compare ATS-GRU with the traditional time series network traffic prediction method (FTS-GRU). To ensure a fair comparison, the two prediction methods used the same number of hidden layers. Figure 5 shows the prediction results of both methods on traffic flow within the same time span. To begin with, the results suggest that the FTS-GRU model exhibits relatively lower accuracy. This is primarily attributed to its tendency to produce predictions that closely align with the average traffic volume for the entire prediction period at each time point. Consequently, this approach may struggle to effectively capture fluctuations and abrupt shifts in traffic flow, potentially leading to inaccuracies in certain scenarios. Conversely, the ATS-GRU model excels in enhancing prediction accuracy by forecasting both the overarching trends and sudden fluctuations in traffic volume. As traffic flow undergoes changes, this model dynamically adapts by reducing the time step as the speed increases. This dynamic adjustment allows the prediction model to more adeptly capture rapid shifts in traffic flow, resulting in more precise prediction outcomes.

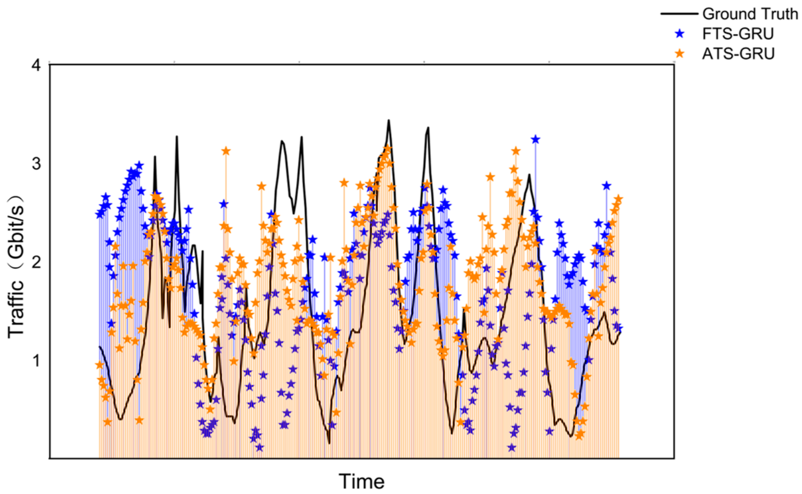


**Figure 5.** Traffic prediction curve obtained with different model.

In addition, we compare the average prediction error of both methods on ten traffic flow profiles using the NRMSE metric, as shown in Figure 6. Consistent with expectations, the ATS-GRU model has significantly lower prediction error than the FTS-GRU model for all traffic flow profiles, thus reinforcing the superior prediction performance of the ATS-GRU model. The difference in error between ATS-GRU and traditional FTS-GRU primarily stems from the differences in the time-stepping size used for prediction. As mentioned earlier in the article, FTS-GRU employs a fixed step size for data sampling to complete predictions. Regardless of whether this step size is large or small, it is challenging to accurately forecast data within each time period because a dataset invariably contains instances of both small and large fluctuations in traffic volume. In contrast, ATS-GRU can precisely monitor traffic fluctuations through error feedback and dynamically adjust the time-stepping size based on these fluctuations. Therefore, it can be confidently stated that the predictive performance of the ATS-GRU model is superior to that of the FTS-GRU model, with errors approaching zero.

Next, the importance and advantages of the ATS method in traffic load prediction are further explained and illustrated through a comparison of the prediction results using three different step sizes based on NLMS, namely FSS, VSS, and ATS.

Figure 7 exhibits the normalized processing time for the three NLMS-based prediction methods. The comparison of different methods' processing time is essential in determining the efficiency of each method. To avoid the reliance of simulations on specific hardware resources, the FSS-NLMS method's running time is set to one. The VSS-NLMS and ATS-NLMS methods' running time is then normalized based on this criterion. The superior computational efficiency of ATS-NLMS is evident in Figure 7, where it demonstrates significantly lower normalized running times compared to FSS-NLMS and VSS-NLMS.

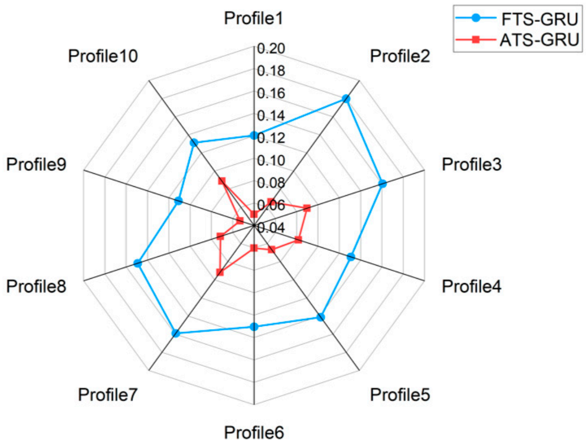


**Figure 6.** Traffic prediction errors obtained with different model.

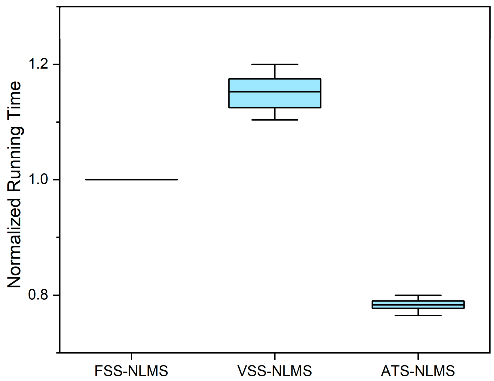


**Figure 7.** Box plot of three algorithms for normalized running time.

Furthermore, ATS-NLMS boasts another significant advantage compared to FSS-NLMS and VSS-NLMS. As shown in Figure 8, the box plots of the prediction errors for the three prediction methods are presented. ATS-NLMS and VSS-NLMS exhibit similar predictive performance, which is significantly better than that of FSS-NLMS, with only a few simulations demonstrating the opposite. Neglecting these isolated simulation points is acceptable because the focus is on the overall effectiveness of the methods. ATS-NLMS stands out with a lower average sampling rate, as depicted in Figures 7 and 8. These figures demonstrate that while FSS-NLMS and VSS-NLMS achieve a time step size of 420 s using the identical simulation configuration, ATS-NLMS surpasses them with a time step size of 371 s. Consequently, ATS-NLMS attains the lowest sampling rate among the three methods, leading to a substantial reduction in sampling overheads.

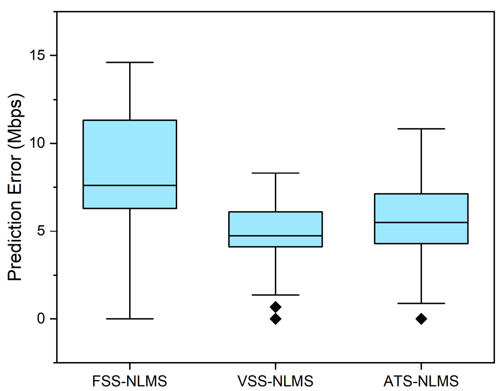


**Figure 8.** Box plot of three algorithms for prediction errors.

## 6. Conclusions

In this paper, we present a network traffic prediction method utilizing machine learning and GRU techniques with adaptive time stepping. In scenarios where the time-stepping size can adapt dynamically based on real-time fluctuations in network traffic, our method demonstrates enhanced accuracy in predicting network traffic compared to conventional approaches that rely on fixed time-stepping sizes. In 6G satellite networks, the ATS-GRU algorithm dynamically adjusts sampling intervals for accurate real-time traffic prediction, even in challenging scenarios like high-speed mobile connections and network congestion. It enhances network management by optimizing resource allocation based on precise predictions, reducing costs, and maintaining network performance, which is crucial for cost-effective broadband satellite communication. The ATS-GRU algorithm not only offers a novel approach to traffic prediction in the realms of communication networks, smart cities, and the Internet of Things, but also has the potential to monitor medical devices and patient health, thereby enhancing the efficiency and quality of healthcare services. Additionally, it can be employed to forecast financial market data flows, financial transactions, and investment trends, facilitating more precise decision making and improved risk management.

**Author Contributions:** Methodology, X.Z.; validation, Y.Z.; writing—original draft, X.Z.; writing—review and editing, P.Y. and X.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by National Key R&D Plan [Grant number: 2021YFB2800805].

**Data Availability Statement:** Data sharing is applicable.

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

## References

1. Kaur, J.; Khan, M.A.; Iftikhar, M.; Imran, M.; Haq, Q.E.U. Machine Learning Techniques for 5G and Beyond. *IEEE Access* **2021**, *9*, 23472–23488. [CrossRef]
2. Chen, C.; Ekici, E. A routing protocol for hierarchical LEO/MEO satellite IP networks. *Wirel. Netw.* **2005**, *11*, 507–521. [CrossRef]
3. Lin, Z.; Lin, M.; Champagne, B.; Zhu, W.-P.; Al-Dhahir, N. Secrecy-Energy Efficient Hybrid Beamforming for Satellite-Terrestrial Integrated Networks. *IEEE Trans. Commun.* **2021**, *69*, 6345–6360. [CrossRef]
4. Lin, Z.; Niu, H.; An, K.; Wang, Y.; Zheng, G.; Chatzinotas, S.; Hu, Y. Refracting RIS-Aided Hybrid Satellite-Terrestrial Relay Networks: Joint Beamforming Design and Optimization. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 3717–3724. [CrossRef]
5. An, K.; Lin, M.; Ouyang, J.; Zhu, W.-P. Secure Transmission in Cognitive Satellite Terrestrial Networks. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 3025–3037. [CrossRef]
6. Chen, Z.; Hu, J.; Chen, X.; Hu, J.; Zheng, X.; Min, G. Computation Offloading and Task Scheduling for DNN-Based Applications in Cloud-Edge Computing. *IEEE Access* **2020**, *8*, 115537–115547. [CrossRef]
7. Khan, P.W.; Abbas, K.; Shaiba, H.; Muthanna, A.; Abuarqoub, A.; Khayyat, M. Energy efficient computation offloading mechanism in multi-server mobile, edge computing—An integer linear optimization approach. *Electronics* **2020**, *9*, 1010. [CrossRef]
8. Song, G.; Chao, M.; Yang, B.; Zheng, Y. TLR: A traffic light-based intelligent routing strategy for NGEO satellite IP networks. *IEEE Trans. Wirel. Commun.* **2014**, *13*, 3380–3393. [CrossRef]
9. Nishiyama, H.; Tada, Y.; Kato, N.; Yoshimura, N.; Toyoshima, M.; Kadowaki, N. Toward optimized traffic distribution for efficient network capacity utilization in two-layered satellite networks. *IEEE Trans. Veh. Technol.* **2013**, *62*, 1303–1313. [CrossRef]
10. Gamvros, I.; Raghavan, S. Multi-period traffic routing in satellite networks. *Eur. J. Oper. Res.* **2012**, *219*, 738–750. [CrossRef]
11. Moscholios, I.D.; Vassilakis, V.G.; Sarigiannidis, P.G.; Sagias, N.C.; Logothetis, M.D. An analytical framework in LEO mobile satellite systems servicing batched Poisson traffic. *IET Commun.* **2018**, *12*, 18–25. [CrossRef]
12. Di, B.; Zhang, H.; Song, L.; Li, Y.; Li, G.Y. Ultra-dense LEO: Integrating terrestrial-satellite networks into 5G and Beyond for Data Offloading. *IEEE Trans. Wirel. Commun.* **2018**, *18*, 47–62. [CrossRef]
13. Li, N.; Hu, L.; Deng, Z.-L.; Su, T.; Liu, J.-W. Research on GRU Neural Network Satellite Traffic Prediction Based on Transfer Learning. *Wirel. Pers. Commun.* **2021**, *118*, 815–827. [CrossRef]
14. Zhao, J.-H.; Wang, M.-X.; Qu, H.; Xie, Z.-Y.; Liu, X. An Adaptive KLMS Traffic Prediction Algorithm for Satellite Network. *Beijing Youdian Daxue Xuebao/J. Beijing Univ. Posts Telecommun.* **2018**, *41*, 51–55. [CrossRef]
15. Liu, Z.; Li, W.; Feng, J.; Zhang, J. Research on Satellite Network Traffic Prediction Based on Improved GRU Neural Network. *Sensors* **2022**, *22*, 8678. [CrossRef] [PubMed]

16. Yang, L.; Gu, X.; Shi, H. A Noval Satellite Network Traffic Prediction Method Based on GCN-GRU. In Proceedings of the 2020 International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, 21–23 October 2020; pp. 718–723. [CrossRef]
17. Zhu, F.; Liu, L.; Lin, T. An LSTM-based Traffic Prediction Algorithm with Attention Mechanism for Satellite Network. In Proceedings of the AIPR 2020: 2020 3rd International Conference on Artificial Intelligence and Pattern Recognition, Xiamen, China, 26–28 June 2020; pp. 205–209.
18. Liu, Z.; Han, J.; Wang, Y.; Li, X.; Chen, S. Performance analysis of routing algorithms in satellite network under node failure scenarios. In Proceedings of the IEEE Global Communications Conference (GLOBECOM'14), Austin, TX, USA, 8–12 December 2014; IEEE: New York, NY, USA, 2014; pp. 2838–2843.
19. Fan, J.; Mu, D.; Liu, Y. Research on Network Traffic Prediction Model Based on Neural Network. In Proceedings of the 2019 2nd International Conference on Information Systems and Computer Aided Education (ICISCAE), Dalian, China, 28–30 September 2019; pp. 554–557. [CrossRef]
20. Organisation for Economic Co-Operation and Development. OECD Digital Economy Outlook 2015. Available online: https://www.broadbandcommission.org/Documents/reports/bb-annualreport2015.pdf (accessed on 11 September 2023).
21. Kawamoto, Y.; Nishiyama, H.; Kato, N.; Kadowaki, N. A traffic distribution technique to minimize packet delivery delay in multilayered satellite networks. *IEEE Trans. Veh. Technol.* **2014**, *62*, 3315–3324. [CrossRef]
22. Wu, Z.; Hu, G.; Jin, F.; Song, Y.; Fu, Y.; Ni, G. A novel routing design in the IP-based GEO/LEO hybrid satellite networks. *Int. J. Satell. Commun. Netw.* **2016**, *35*, 179–199. [CrossRef]
23. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In NIPS. 2012. Available online: https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html (accessed on 11 September 2023).
24. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
25. Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bougares, F.; Schwenk, H.; Bahdanau, D.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
26. Wu, L.; Kong, C.; Hao, X.; Chen, W. A short-term load forecasting method based on GRU-CNN hybrid neural network model. *Math. Probl. Eng.* **2020**, *2020*, 1428104. [CrossRef]