

Deep Recurrent Neural Networks for Ionospheric Variations Estimation Using GNSS Measurements

Maria Kaselimi¹, Athanasios Voulodimos², *Member, IEEE*, Nikolaos Doulamis¹,
Anastasios Doulamis¹, *Member, IEEE*, and Demitris Delikaraoglou¹

Abstract—Modeling ionospheric variability throughout a proper total electron content (TEC) parameter estimation is a demanding, however, crucial, process for achieving better accuracy and rapid convergence in precise point positioning (PPP). In particular, the single-frequency PPP (SF-PPP) method lacks accuracy due to the difficulty of dealing adequately with the ionospheric error sources. In order to apply ionosphere corrections in techniques, such as SF-PPP, external information of global ionosphere maps (GIMs) is crucial. In this article, we propose a deep learning model to efficiently predict TEC values and to replace the GIM-derived data that inherently have a global character, with equal or better in accuracy regional ones. The proposed model is suitable for predicting the ionosphere delay at different locations of receiver stations. The model is tested during different periods of time, under different solar and geomagnetic conditions and for stations in various latitudes, providing robust estimations of the ionospheric activity at the regional level. Our proposed model is a hybrid model comprising of a 1-D convolutional layer used for the optimal feature extraction and stacked recurrent layers used for temporal time series modeling. Thus, the model achieves good performance in TEC modeling compared to other state-of-the-art methods.

Index Terms—3-D tensor, global navigation satellite system, ionospheric variability, recurrent neural network (RNN), total electron content (TEC).

I. INTRODUCTION

THE ionosphere is typically defined as that part of the Earth's upper atmosphere with a sufficient concentration of free electrons affecting the propagation of electromagnetic waves. The electrons density depends on the time of the day and the Sun's activity, the atmospheric density profile, the geographic location, and the magnitude and orientation of the Earth's magnetic field [2]. The total electron content (TEC), defined as the integral of the electron density over a signal path, is often used to describe ionosphere variability [3], [4].

Manuscript received September 28, 2020; revised January 18, 2021, April 15, 2021, and May 12, 2021; accepted May 12, 2021. Date of publication July 1, 2021; date of current version January 12, 2022. This work was supported by the European Health and Digital Executive Agency (HADEA) under the powers delegated by the European Commission through the Horizon 2020 Program "HEALTHier Cities through Blue-Green Regenerative Technologies: the HEART Approach" under Grant Agreement 945105. (*Corresponding author: Maria Kaselimi.*)

Maria Kaselimi, Nikolaos Doulamis, Anastasios Doulamis, and Demitris Delikaraoglou are with the School of Rural, Surveying and Geoinformatics Engineering, National Technical University of Athens, 15780 Athens, Greece (e-mail: mkaselimi@mail.ntua.gr; ndoulam@cs.ntua.gr; adoulam@cs.ntua.gr; ddeli@mail.ntua.gr).

Athanasios Voulodimos is with the Department of Informatics and Computer Engineering, University of West Attica, 12243 Psachna, Greece (e-mail: avoulod@uniwa.gr).

Digital Object Identifier 10.1109/TGRS.2021.3090856

Since radio wave signals pass through the electrons of the ionosphere, the signal velocity and the ray path change [5].

Consequently, the transmitted signals from the Global Navigation Satellite Systems (GNSS) are directly affected by the ionospheric variations, causing delays [6]. These delays depend on the signal frequency and the electron density along the transmission path. Hence, ionospheric variability introduces an additional error source in GNSS positioning [7]. The use of multiple navigation signals of distinct center frequency transmitted from the same GNSS satellite allows direct estimation of these ionospheric delays. Exploiting the fact that different signal frequencies are affected differently by the ionosphere, an appropriate processing strategy of multiple-frequency GNSS signals, eliminates the ionospheric error [7]. Contrary to multifrequency GNSS receivers, real-time (RT) single-frequency (SF) positioning with a low-cost receiver has received increasing attention in recent years due to its large amount of possible applications. However, in this case, one major challenge is the effective mitigation of these ionospheric delays [8]. RT-SF-standard point positioning (SPP)/precise point positioning (PPP) techniques use ionospheric vertical TEC (VTEC) products released by the International GNSS Service (IGS) RT service [9] to eliminate the ionospheric error and apply corrections to the model as external parameters [1]. However, these ionospheric VTEC products have global coverage.

Many researchers investigate the ionospheric variation, applying regional models. Linear or nonlinear filters are employed to model the TEC values at the regional level. As far as linear modeling is concerned, autoregressive (AR) or AR moving average (ARMA) filters derived from the time series signal processing research community are applied [10], [11]. These filters are linear regressions in the time domain, and thus, they cannot capture the complex and highly nonlinear phenomena that occurred in the ionosphere.

There are various useful indicators for TEC modeling. These parameters (indicators) related to geomagnetic and solar activity conditions are given as follows.

Geomagnetic Indices: Here, the planetary Kp-index and the Ap-index are used, while the disturbance storm-time Dst-index is related with ionosphere short-term changes from several hours to several days [12].

Solar Indices: The sunspot number (SSN) and the solar radio flux F10.7 better suit long-term variations of the order of months (27-day solar rotation) [13].

In this article, we overcome the aforementioned limitation of the linear filters by constructing robust nonlinear regional models for ionospheric variability modeling, applying as external ionosphere corrections. In particular, we propose a spatiotemporal deep learning architecture for electron density modeling, combining a convolutional neural network (CNN) to capture the spatial variability of TEC, and a gated recurrent unit (GRU) for temporal variability modeling. The proposed deep recurrent architecture successfully models ionosphere conditions and estimate the ionospheric delays on the GNSS satellite signals, treating ionosphere variations as a nonlinear time series regression problem.

A. Related Work

An AR model is a common linear regression approach for modeling of TEC values at a regional level [14], [15]. Such architectures model the TEC values based on previously measured electron density data. The limitation of an AR filter is that no external measurable parameters are allowed to be utilized by the model. Thus, ARMA filters have been investigated for TEC modeling [10], [16]. The main difference between an AR and ARMA process is that the first (i.e., AR) models a time series in terms of its own lags (previous observable measurements), while an ARMA filter includes two parts: an AR where previous observations affect the output and the moving average (MA) part where external observations trigger the output. In this context, the work of [11] proposes an adaptive AR process to capture the time variations of the electron density. In addition, the work of [17] combines wavelet transform with AR models for ionosphere variations estimation. However, the main common drawback of these linear models is that they fail to capture the abrupt changes in electron density values due to the linear assumptions made. In addition, linear regressions are not adequate to model abrupt changes existing in TEC signals.

Recently, and mainly due to the advances in artificial intelligence (AI) research, many efforts are utilized to develop TEC models using nonlinear regression architectures based on machine learning algorithms. In this context, a widely used model is the feedforward neural network, consisting of interconnected artificial neurons capable of modeling nonlinear input-output relationships [18]. The feedforward neural network examples for modeling electron density signals are the works of [19], [20], and [21]. In fact, feedforward neural networks are capable of approximating nonlinear ARMA relationships, therefore improving the performance in TEC modeling. Nonlinear ARMA filters with recursive capabilities have been also proposed in [22] and [23]. Other works in this field apply radial basic function (RBF) models [24], with advance nonlinear interpolation capabilities, or support vector machines (SVMs) [25]. The latter are supervised learning paradigms for data classification and regression.

The main limitations of the aforementioned nonlinear regression models are that they present convergence instabilities, especially when a large number of neurons are employed in the network. In addition, there are no recurrent feedback mechanisms among the artificial neurons. Therefore, such

filters fail to approximate temporal dependencies and high abrupt changes in TEC time series values with high precision accuracy.

For this reason, recently, deep machine learning has been proposed as an alternative paradigm for regression and classification [26]. Deep learning incorporates multiple hidden neurons and applies advanced learning algorithms, such as input compression and dimensionality data reduction, to handle the computational issues arising when a large number of neurons are considered. Recurrent neural networks (RNNs) are a class of networks allowing connections (feedback) between the nodes (neurons) in order to model the temporal behaviors of a time series signal [27]. Thus, RNNs are capable of handling the time dependencies in TEC modeling. However, RNNs fail to approximate more complex temporal dependencies, presenting also computational issues in computing the gradient during the learning process, especially when a large number of neurons are employed (the so-called vanishing gradient problem [28]).

To address these limitations and simultaneously to retain the advantages of deep learning in approximating temporal dependencies, long short-term memory (LSTM) architectures have been recently proposed [29]. In the context of TEC modeling, LSTM networks memorize temporal correlations of the TEC signals, therefore providing better modeling capabilities [30]–[32].

Although the aforementioned architectures are good modeling approximators of TEC, they mainly focus on the temporal modeling dimension, which is of estimating TEC changes between different temporal epochs. However, a TEC time series signal also depends on the geographic location, and therefore, different weights should be assigned to the model for different station latitudes. This so-called *spatial variability* cannot be modeled through the traditional LSTM deep learning architectures. In addition, LSTMs have more trainable parameters (e.g., weights) compared to the traditional RNN models. Consequently, they give us, on the one hand, more controllability and, thus, better results but with the cost, on the other hand, of more complexity and the need for large amount of training (annotated) data.

B. Contribution

To overcome the aforementioned issues, in this article, we introduce a new *spatiotemporal* deep learning paradigm for TEC modeling. The proposed method combines two deep learning structures: a *CNN* and a *GRU*. These two structures are combined together, forming a common trainable model.

In particular, the Earth's ionosphere shows marked variations with latitude, longitude, universal time, season, solar, and geomagnetic activity. Thus, our proposed model feeds all these variables as inputs to the model and relates them under a nonlinear relationship. TEC is characterized by high complexity, and it is *space- and time-varying*. In order to capture the spatiotemporal behavior of TEC, in our proposed model, measurements of every satellite visible from the observing location (ground station) are taken into account. This means that the model's input data form a spatial–temporal 3-D tensor.

The tensor consists of several time series data corresponding to different single satellite visible from the station over a time window period.

The CNN captures the spatial variability of the TEC values assigning different learnable weights to the following GRU, taking into account all the visible time series data from a ground station. The main operational unit of a CNN architecture is the convolutional kernels, units able to model with high-efficiency spatial signal properties under complex nonlinear relationships [26]. In other words, the purpose of the CNN is to appropriately adjust the model weights to better capture the particularities of each station.

On the other hand, the GRUs model the temporal variability of the ionosphere variability. GRUs are similar deep learning structures with LSTM, however requiring fewer parameters. Therefore, they retain the temporal-dependent capabilities of the LSTM, but they advance in terms of complexity (and, therefore, convergence) and the need for smaller training (annotated) datasets. For this reason, GRUs have been selected in this article for modeling the temporal ionospheric variability. In other words, our model leverages the capability of the CNNs to optimally approximate the spatial particularities of each station, based on a set of weights, along with GRUs able to learn the temporal dependencies of the TEC values.

Another contribution of this article is that our nonlinear model is based on the so-called *sequence-to-sequence* (Seq2seq) modeling framework; a family of machine learning algorithms to transform one sequence into another sequence. Seq2seq methods have been originated by Google engine for text-/speech-language translation [33], but they have recently been adopted in solving complex time series problems in the power energy domain [34]. The Seq2seq structure permits modeling of the ionosphere values for all satellites in view at every epoch [35]. Given that there are multiple input and output time steps, this problem is referred to as “*many-to-many*” sequence modeling problem. The Seq2seq modeling approach means that we are able to model not only the electron density values at the next epoch but a sequence of successive epochs.

The remainder of this article is structured as follows. Section II presents a brief background on GNSS and ionospheric variability before providing a formulation of the problem discussed. Section III describes the proposed spatiotemporal deep learning TEC model. Section IV describes the proposed model architecture that combines convolutional and recurrent layers together in an optimal structure. In Section V, an extensive experimental evaluation of the discussed methods is provided, while Section VI closes this article with a summary of findings.

II. IONOSPHERE MODELING PROBLEM FORMULATION BASED ON GNSS OBSERVATIONS

A. GNSS Observations Preprocessing

As we have previously stated, our goal is to construct robust regional models of the ionosphere variability, applied in SF-PPP methods as external ionosphere correction information. During the training of these models, the ground-truth

TEC values are taken into consideration. The workflow is given as follows: we first apply a dual-frequency undifferenced and unconstrained PPP model to estimate slant TEC (STEC) values. These values are then separated from satellite and receiver differential code biases (DCBs). Then, having pure STEC values, we convert them to VTEC ones. These VTEC ground-truth (labeled) values are combined with solar and geomagnetic indicators to construct the TEC model using a supervised learning framework [26], as explained in Section II-B. For this reason, in the following, we describe the methodology adopted for VTEC values estimation.

The code P and phase ϕ observations in a given frequency band f_i between a receiver r and a GNSS satellite s are written as [36]

$$P_{(fi)r}^s = \rho_r^s + c \cdot (dt_r + dt^s) + d_{(fi)r} - d_{fi}^s + T^s + I_{(fi)}^s + \epsilon_{P_{(fi)}}^s \quad (1)$$

$$\phi_{(fi)r}^s = \rho_r^s + c \cdot (dt_r + dt^s) + \delta_{(fi)r} - \delta_{fi}^s + T^s - I_{(fi)}^s + \lambda N_{(fi)}^s + \epsilon_{\phi_{(fi)}}^s \quad (2)$$

where $P_{(fi)r}^s$ and $\phi_{(fi)r}^s$ denote pseudorange and carrier phase observables, respectively; ρ_r^s is the geometric distance between the receiver to the satellite; dt_r and dt^s are the receiver and satellite clock offsets, respectively; $d_{(fi)r}$ is the frequency-dependent receiver uncalibrated code delay (UCD), while d_{fi}^s is the frequency dependent satellite UCD; T^s is troposphere delay; $I_{(fi)}^s$ is the line of sight (LOS) ionospheric delay on the frequency fi ; $\delta_{(fi)r}$ and δ_{fi}^s are the frequency-dependent receiver and satellite uncalibrated phase delay, respectively; $N_{(fi)}^s$ is the phase ambiguity; and $\epsilon_{P_{(fi)}}^s$ and $\epsilon_{\phi_{(fi)}}^s$ are the sum of measurement noise and multipath error for pseudorange and carrier phase observations.

In the case of dual-frequency GPS observations and assuming the frequencies f_1 and f_2 noted as “1” and “2,” respectively, (1) is written as

$$P_{1r}^G = \rho_r^G + c \cdot (dt_r + dt^G) + d_{1r} - d_1^G + T^G + I_1^G + \epsilon_{P_1}^G \quad (3)$$

$$P_{2r}^G = \rho_r^G + c \cdot (dt_r + dt^G) + d_{2r} - d_2^G + T^G + I_2^G + \epsilon_{P_2}^G. \quad (4)$$

The code biases are commonly referred as DCBs $\text{DCB} = \text{DCB}_{P_1/P_2} = d_{f_1} - d_{f_2}$, and given that $\gamma_2 = f_1^2/f_2^2$, we have

$$d_{f_1} = d_{\text{IF}} + 1/(1 - \gamma_2) \cdot \text{DCB} \quad \text{and} \\ d_{f_2} = d_{\text{IF}} + \gamma_2/(1 - \gamma_2) \cdot \text{DCB}. \quad (5)$$

Then,

$$d_{f_1r} - d_{f_1}^G = d_{(\text{IF})r} - d_{\text{IF}}^G + \frac{1}{(1 - \gamma_2)} \cdot (\text{DCB}_r - \text{DCB}^s) \quad (6)$$

$$d_{2r} - d_2^G = d_{(\text{IF})r} - d_{\text{IF}}^G + \frac{\gamma_2}{(1 - \gamma_2)} \cdot (\text{DCB}_r - \text{DCB}^s). \quad (7)$$

The term \tilde{I}_1 is computed as

$$\tilde{I}_1 = I_1 - \frac{1}{(1 - \gamma_2)} \text{DCB}^s + \frac{1}{(1 - \gamma_2)} \text{DCB}_r. \quad (8)$$

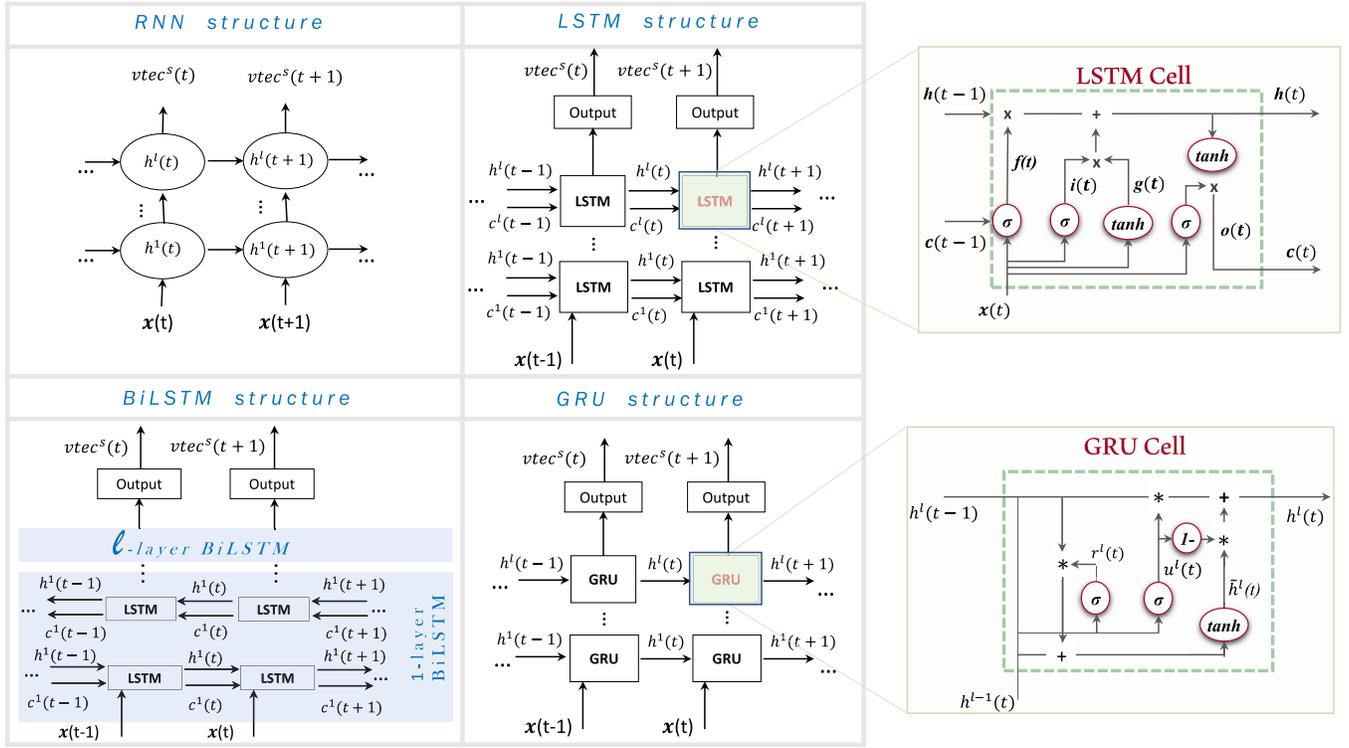


Fig. 1. Stacked RNN-based structures for TEC modeling.

The uncombined PPP (UPPP) model computes the ionosphere delay as unknown parameter, in contrast to the traditional ionosphere-free (IF) model that combines multiple frequency observations to eliminate the ionospheric error. Our model is ionosphere-constrained dual-frequency PPP model that estimates the STEC values. These values are then separated from satellite and receiver DCBs, using the products of IGS service [37]. Finally, the VTEC values, called with the variable $vtec$ in this article, are estimated as

$$vtec = \frac{1}{MF_I} \cdot \left(\tilde{I}_1 + \frac{1}{(1 - \gamma_2)} DCB^s - \frac{1}{(1 - \gamma_2)} DCB_r \right). \quad (9)$$

The above equation evaluates the ground-truth VTEC values. Then, the STEC, denoted with the variable $stec$, is converted into $vtec$ through the mapping function MF_I [38]

$$MF_I = \frac{stec}{vtec} = \frac{1}{\left(1 - \left(\frac{R_e}{R_e + h_s} \cos\theta \right)^2 \right)^{1/2}} \quad (10)$$

where R_e is the mean Earth's radius, θ is the satellite's elevation angle, and h is the height of the ionospheric layer and usually has been taken about 350 km.

The GNSS preprocessing is implemented using the GAMP [36] GNSS processing software. Further details about the experimental setup are provided in Table I.

B. Nonlinear Ionosphere TEC Modeling

In Section II-A, we have estimated the ground-truth values of VTEC through (9). A typical model assumes that the ionosphere is a thin shell above the Earth, located near the mean altitude of maximum TEC (approximately 350 km). The intersection between a signal's LOS and this shell is

TABLE I
STATIC PPP EXPERIMENTAL SETUP

Options	Settings
Constellation	GPS
Positioning mode	static PPP
Frequencies	L1, L2
Sampling rate	1 min
Elevation mask	7°
Differential code bias (DCB)	MGEX DCB products for PPP
Receiver and Satellite antenna	corrected with igs14.atx
Satellite orbits and clock offsets	Wuhan University products
Phase wind-up	corrected
Sagnac effect, relativistic effect	corrected with IGS absolute
Station reference coordinates	IGS SINEX solutions

called the ionospheric pierce point (IPP). In our model, we use as input the IPP points coordinates, noted by x_ϕ and x_λ . In addition, the information of daily time hour x_{dt} has been incorporated into the model to boost its performance. Let us denote as $vtec^s(t)$ the value of the VTEC at a time instance t for a visible satellite s . Then, we have that

$$\begin{aligned} vtec^s(t) &= f(\mathbf{x}(t)) + e(t) \\ &= f(x_\phi(t), x_\lambda(t), x_{SSN}(t), x_{DST}(t), x_{F10.7}(t), \\ &\quad x_{Ap}(t), x_{Kp}(t), x_{dt}(t)) + e(t). \end{aligned} \quad (11)$$

In (11) $f(\cdot)$ refers to the nonlinear relationship between the VTEC values and the inputs of solar and geomagnetic indices. It is clear that this nonlinear relationship is actually unknown, and therefore, it is approximated by the proposed spatiotemporal deep learning model. The model parameters (weights) of the model are used to approximate $f(\cdot)$. These weights are estimated through the application of a supervised learning methodology [26]. In (11), variables x_{SSN} and $x_{F10.7}$ refer to the input solar indices, and x_{DST} , x_{Kp} , and x_{Ap} are the

geomagnetic indices used as input variables of the proposed nonlinear model (see Section I).

III. SEQUENCE-TO-SEQUENCE SPATIOTEMPORAL AI FOR TEC MODELING

As we have previously stated, a spatiotemporal deep learning model is used, in this article, to model the ionospheric variability. The proposed deep learning model consists of a convolutional layer (see Section III-A) and stacked GRU recurrent layers (see Section III-B).

A. Spatial Variability Modeling: The Convolutional Neural Network Layer

The purpose of the convolutional layer is to encode the spatial variability of the input signals by assigning different weights to the model depending on the station latitude. In other words, the output of the convolutional layer is to transform the input signals into a vector representation more suitable for TEC modeling. The weights of the convolutional layer are learnable during the training phase and each input contributes in a different way to the model, for different ground stations. Let us denote as $\mathcal{I}_{\text{input}}$ the input sequence of data which are feeding to the convolutional layer. Then, the CNN transforms these inputs $\mathcal{I}_{\text{input}}$ to an encoded vector $f_{\mathcal{M}}$ more suitable for TEC modeling

$$f_{\mathcal{M}} \sim \text{Conv}_{\mathcal{M}}(\mathcal{I}_{\text{input}})$$

with

$$\mathcal{I}_{\text{input}}(t) = [x_{\phi} \ x_{\lambda} \ x_{\text{ssn}} \ x_{\text{Dst}} \ x_{Ap} \ x_{F107} \ x_{Kp} \ x_{dt}]^T. \quad (12)$$

Then, the encoded vector $f_{\mathcal{M}}$ feeds the GRU. The convolutional layer architecture and its position with respect to the whole deep learning model are shown in Fig. 2.

B. Temporal Variability Modeling: The Gated Recurrent Unit Layer

As we have previously stated, GRUs are simpler forms of LSTM models having fewer gates than the LSTM memory cell [39]. In addition, the LSTM networks are deep learning extensions of the RNN model, better modeling the nonlinear attributes of a time series signal [31]. Therefore, before describing the GRU used for modeling the temporal variability of TEC, we discuss the structure of the RNNs and their stacked version and LSTM networks

A stacked recurrent model is an extension to the traditional one consisting of several recurrent layers one stacked over the other [40]. This is presented in Fig. 1. A stacked model has two main types of operation; *the in-depth* and *the temporal operation*. The in-depth operation implies that the response of one layer is propagated as input to the next layer. Instead, the temporal operation assumes that inputs at previous time instances trigger the current unit. The stacked approach adopted in this article is applied for all recurrent structures, that is, the RNN, the LSTM and its bidirectional mode, and the GRU.

1) *Stacked Recurrent Neural Networks*: Recurrent models are powerful tools for time series modeling. The main operational unit of an RNN is an artificial neuron, approximating a nonlinear operation of an inner product of the network weights (parameters) and output responses of other neurons or model input vectors. The difference of an RNN with a traditional neural network model is that, in an RNN, each neuron is also triggered from the response of other neurons at previous time instances, allowing the modeling of temporal dependencies. These neurons are also called hidden states since they are located between the input vector and the output of the model. In particular, the response of an artificial neuron is given by

$$h^l(t) = \tanh(\mathbf{W}_l^T h^l(t-1) + \mathbf{U}_l^T h^{l-1}(t)), \quad l > 1 \quad (13)$$

where $\tanh(\cdot)$ is the hyperbolic tangent function, referring to the nonlinear operational unit of a single neuron of the RNN. Variables \mathbf{W} and \mathbf{U} are the learnable weight parameters of the model. The $h^l(t)$ is the response of a neuron at the l th level of the network at a time instance t . It should be mentioned $h^0(t) \equiv f_{\mathcal{M}}$ coincides with the output response of the convolutional layer, that is, the vector $f_{\mathcal{M}}$ of (12), instead of the previous hidden state $h^{l-1}(t)$.

Once the top-level hidden state is computed, the estimate of the output $\widehat{\text{vtec}}_s$ is obtained using (see Fig. 1)

$$\widehat{\text{vtec}}_s(t) = \tanh(\mathbf{V}^T h^l(t)) \quad (14)$$

where \mathbf{V} is a matrix corresponding to the output learnable parameters (weights) of the model.

2) *LSTM Structure*: LSTM is a special kind of the traditional RNN structure, where each node in the hidden layer is replaced by a more complex structure, called memory cell, instead of RNN's single neuron [41]. The core structure of a single memory cell is presented in Fig. 1. The memory cell contains three different components: 1) the forget gate $f(t)$; 2) the input gate $i(t)$; 3) the cell candidate $g(t)$; and 4) the output gate $o(t)$. For each component, a nonlinear relation to the inner product between the input vectors and respective weights is applied during the training process. In some of the components, the sigmoid function $\sigma(\cdot)$ is applied, while, in others, the hyperbolic tangent function $\tanh(\cdot)$ is used. The forget gate $f(t)$ keeps unnecessary information out of the memory cell, thus separating the worth-remembering information from the useless one [42]. The input gate $i(t)$ regulates whether the information is relevant enough to be applied in future steps for the accurate estimation of TEC values. The cell candidate $g(t)$ activates appropriately the respective state (true or false output from the \tanh activation). The output gate $o(t)$ decides if the response of the current memory cell is "significant enough" to contribute to the next cell.

Regarding stacked LSTMs, the additional LSTM layers can recombine the learned representation from prior layers and create new representations at high levels of abstraction. Stacked LSTMs or deep LSTMs were introduced by Graves [43] in their application of LSTMs to speech recognition, beating a

benchmark on a challenging standard problem

$$\begin{bmatrix} i^l(t) \\ f^l(t) \\ o^l(t) \\ g^l(t) \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} \mathbf{W}_l \begin{bmatrix} h^l(t-1) \\ h^{l-1}(t) \end{bmatrix} + \begin{bmatrix} b_i^l \\ b_f^l \\ b_o^l \\ b_g^l \end{bmatrix}. \quad (15)$$

When $l = 1$, the state is computed using the encoded vector f_M (see Section III-A) instead of $h^{l-1}(t)$.

The cell state at time step t is given by

$$c^l(t) = f^l(t) \odot c^l(t-1) + i^l(t) \odot g^l(t) \quad (16)$$

where \odot denotes the Hadamard product (elementwise multiplication of vectors).

The hidden state at time step t is given by

$$h^l(t) = o^l(t) \odot \sigma(c^l(t)) \quad (17)$$

where σ denotes the state activation function.

3) *Bidirectional LSTM*: The stacked bidirectional LSTM (BILSTM) has a similar structure with the LSTM model with the difference that it allows bidirectional data processing. Therefore, the operational units of a bidirectional LSTM model are defined by two parts: the *forward* and *backward* parts. As far as the forward part is concerned, we have the following equations:

$$\underline{c}^l(t) = \underline{f}^l(t) \odot \underline{c}^l(t+1) + \underline{i}^l(t) \odot \underline{g}^l(t) \quad (18)$$

$$\underline{h}^l(t) = \underline{o}^l(t) \odot \sigma(\underline{c}^l(t)) \quad (19)$$

and backward

$$\underline{c}^l(t) = \underline{f}^l(t) \odot \underline{c}^l(t-1) + \underline{i}^l(t) \odot \underline{g}^l(t) \quad (20)$$

$$\underline{h}^l(t) = \underline{o}^l(t) \odot \sigma(\underline{c}^l(t)). \quad (21)$$

4) *Stacked GRU Architecture*: The GRU is simpler form of the LSTM unit, having two control gates: the *reset* gate $r^l(t)$ and the *update* gate $u^l(t)$ (see Fig. 1). The reset gate $r^l(t)$ is responsible for determining how much of information to forget. The update gate $u^l(t)$ is responsible for determining the worth-remembering information of the previous states that should be forwarded to the next state. Therefore, the gates $r^l(t)$ and $u^l(t)$ are related with the hidden states $h^l(t)$ and $h^l(t-1)$ as follows:

$$\begin{bmatrix} u^l(t) \\ r^l(t) \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \end{bmatrix} \mathbf{W}_l \begin{bmatrix} h^l(t-1) \\ h^{l-1}(t) \end{bmatrix} + \begin{bmatrix} b_u^l \\ b_r^l \end{bmatrix}. \quad (22)$$

In (22), σ is the sigmoid function, and b_u^l and b_r^l are the respective biases of each component for the GRU. Variables \mathbf{W} and \mathbf{U} are the transition matrices of the l th GRU.

In stacked GRU configuration, a recursive approach is considered as regards the operation of each GRU. A new memory state, denoted as $\tilde{h}^l(t)$, acts as the consolidation of the hidden state of the previous layer $h^{l-1}(t)$ and the previous hidden state $h^l(t-1)$ of the current layer. The consolidated hidden state $\tilde{h}^l(t)$ is given by

$$\tilde{h}^l(t) = \tanh(r^l(t)\mathbf{U}h^l(t-1) + \mathbf{W}h^{l-1}(t)). \quad (23)$$

In (23), function $\tanh(\cdot)$ refers to the hyperbolic tangent relationship. Equation (23) means that the consolidated state is related with the output of the hidden state $h^l(t-1)$ at the time instance $t-1$ and the output of the previous hidden layer $h^{l-1}(t)$ at the time instance t . Using the values of the consolidated state $\tilde{h}^l(t)$ and the values of the update gate $u^l(t)$, the value of the hidden state of the l th GRU element is estimated

$$h^l(t) = (1 - u^l(t))\tilde{h}^l(t) + u^l(t)h^l(t-1). \quad (24)$$

In more detail, the GRU-related abovementioned operations are illustrated in Fig. 1.

IV. PROPOSED CNN-GRU ARCHITECTURE FOR VTEC MODELING

A. Implementation Details

Fig. 2 illustrates the CNN-GRU architecture. The network configuration setup consists of: 1) the input layer; 2) a 1-D convolutional layer; 3) two stacked recurrent GRUs; and 4) two dense layers (Fig. 2). The GRU accepts a 3-D tensor input. The number of kernels of the convolutional layer is 60 with a size of 5. The first GRU layer consists of 90 filters with a size of 1×5 , while the second layer consists of 180 filters of the same size. The output is the sequence of VTEC values.

B. Evaluation Metrics

Here, we present the metrics used for model evaluation and comparisons with other linear and nonlinear approximators.

For each satellite s_i , visible in ground station, a sequence of VTEC values is produced using the proposed CNN-GRU deep learning model. Then, we compute the absolute difference between the ground-truth $\text{vtec}^{s_i}(t)$ value, as obtained from the GAMP software, and the estimated from our model $\widehat{\text{vtec}}^{s_i}(t)$, that is, $\text{dvtec}^{s_i}(t) = |\text{vtec}^{s_i}(t) - \widehat{\text{vtec}}^{s_i}(t)|$, for a time instance t . Based on the values of $\text{dvtec}^{s_i}(t)$, the following metrics are considered:

$$\text{mae} = \frac{1}{S} \sum_{s_i=1}^S \text{mae}^{s_i} = \frac{1}{S} \sum_{s_i=1}^S \left(\frac{\sum_{t=1}^T \text{dvtec}^{s_i}(t)}{T} \right) \quad (25)$$

$$\text{mse} = \frac{1}{S} \sum_{s_i=1}^S \text{mse}^{s_i} = \frac{1}{S} \sum_{s_i=1}^S \left(\frac{\sum_{t=1}^T [\text{dvtec}^{s_i}(t)]^2}{T-1} \right) \quad (26)$$

$$\text{min} = \frac{1}{S} \sum_{s_i=1}^S \min_{\forall t \in T} \text{dvtec}^{s_i}(t) \quad (27)$$

$$\text{max} = \frac{1}{S} \sum_{s_i=1}^S \max_{\forall t \in T} \text{dvtec}^{s_i}(t). \quad (28)$$

In the aforementioned equations: 1) mae refers to mean absolute error (mae) for all satellite s_i per ground station; 2) mse to the respective mean square error; and 3) min (max) to the average minimum (maximum) of $\text{dvtec}^{s_i}(t)$ for all visible satellites. Variable T denotes the time period over which the evaluation takes place.

Another evaluation metric used in this article is the percentiles values of (50th, 68th, and 95th) of the $\text{dvtec}(t)$

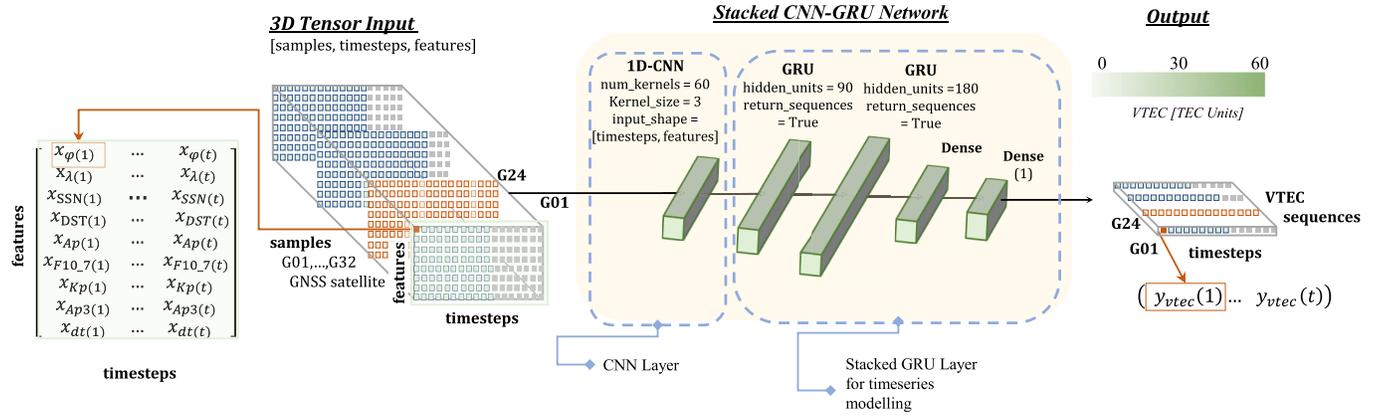


Fig. 2. Proposed model CNN-GRU architecture, consisting of two parts: the CNN for spatial variability modeling and the GRU for temporal variability modeling.

TABLE II
SELECTED SITES FROM THE IGS NETWORK

Site	Latitude (°)	Longitude (°)	Height (m)	Country
graz	47.067	15.493	538.3	Austria
iqal	63.755	-68.510	91.7	Canada
mal2	-2.996	40.194	0.0	Uganda
qaq1	60.715	-46.047	110.4	Greenland
ramo	30.597	34.763	893.1	Israel
tixi	71.634	128.866	46.9	Russia

error distribution per ground station. The error $dv_{tec}(t) = |v_{tec}(t) - \widehat{v_{tec}}(t)|$ is defined as the absolute different between the estimated VTEC value and the ground truth for a station. A 95th percentile quantity means that the values 95% of the VTEC errors are contained, after having sorted all errors in ascending order. Reflecting the error by percentiles is better than by simple mae.

V. EXPERIMENTAL EVALUATION

The experimental setup covers various aspects of the ionosphere phenomenon in order to accurately evaluate the performance of our proposed model. Thus, in our experimental evaluation, we have included stations installed in various latitudes and longitudes at a worldwide level (see Table II). Also, we have included data from different years ranging from 2014 to 2018 to create robust training, validation, and test sets. We have chosen different years under different solar and geomagnetic activities, and also, we have tested different months in order to evaluate our algorithm under different weather and seasonal conditions.

We highlight that our proposed model is not a “pure” GRU model but a hybrid model comprising of convolutional and stacked recurrent GRU layers: *the proposed spatiotemporal deep learning paradigm*. We experimentally validate the superiority of the proposed CNN-GRU model for TEC modeling in comparison to other deep learning networks (LSTM [31], [44], BiLSTM [30], and RNN [45] models) and state-of-the-art techniques for TEC modeling (AR [11] and ARMA [11] models). Also, the proposed method is compared with the conventional global ionosphere map (GIM) and IRI2016 global ionosphere models.

A. Data Preprocessing and Experiment Setup

This section describes the preprocessing approach used to extract the recurrent model inputs and outputs based on observables from the global IGS network of permanent GNSS stations. The necessary observation, navigation, precise orbit and clock, IGS ANTEX (igs14.atx), IGS SINEX, ocean tide loading coefficients, and DCBs are fed into the GAMP software for static PPP processing (see Table I). As described in Section II-A, the GNSS observations are preprocessed to estimate the VTEC values. GNSS observables for the years 2014 to 2018 were processed with 1-min data granularity for the selected ground stations (see Table II).

As far as the deep learning models are concerned, they have been trained and deployed using the Python Tensorflow and Keras libraries. The proposed CNN-GRU and the compared deep learning models have been trained using the adaptive moment estimation optimization algorithm (ADAM) [46] with a learning rate of 10^{-4} . The model weights are updated using a minibatch size of 28 samples at each training iteration. We set the maximum number of epochs for training to 200, that is, the maximum number of training cycles of the network. In our experiments, we use a dataset of the years 2014–2018 (a five-year dataset). This dataset is divided into three subsets: *the Training, the Validation, and the Test Set*. The training set is used for the estimation of the model parameters (weights) during the learning process. The validation set assesses the performance of the model during the learning process on a set different than the one used for training to avoid overfitting. Finally, the test set evaluates the model on data that have not been used during training. In our experiments, the training set consists of 70% of the total available data, while both the validation and the test sets consist of the remaining 15% of the total data. Both training and validation sets cover the period between the years 2014–mid-2018. Data from the second half of 2018 are used as test data.

The AR and ARMA models [11] (see Section I-A) are implemented in Python using the statsmodels library. The order of the AR part for both AR and ARMA is selected to be 60, while the order of the MA part equals 5 in the case of the ARMA model.

TABLE III

PERFORMANCE METRICS (MAE, MSE, MIN, AND MAX) FOR VTEC MODELING FOR SIX SELECTED STATIONS (GRAZ, IQAL, MAL2, QAL1, RAMO, AND TIXI) AMONG THE RNN-BASED METHODS AND TWO OTHER COMMONLY USED METHODS (ARMA AND AR) FOR OCTOBER 9, 2018. MAE AND MSE METRICS ARE THE AVERAGE VALUES OF THE INDIVIDUAL PRN METRICS IN EVERY STATION, WHEREAS MIN AND MAX ARE THE MINIMUM AND MAXIMUM MAE VALUES OBSERVED FOR THE INDIVIDUAL PRNS. THE BEST VALUES ARE PRESENTED IN BOLD

	graz				iqal				mal2			
	mae (TECU)	mse (TECU ²)	min (TECU)	max (TECU)	mae (TECU)	mse (TECU ²)	min (TECU)	max (TECU)	mae (TECU)	mse (TECU ²)	min (TECU)	max (TECU)
proposed	0.714	0.903	0.290	1.961	0.971	1.591	0.390	1.722	1.814	5.878	0.472	3.301
BILSTM [30]	0.824	1.083	0.276	1.968	1.096	2.118	0.457	2.998	2.175	7.814	0.487	3.986
LSTM [44], [31]	0.836	1.304	0.421	1.745	0.940	1.594	0.523	2.411	2.166	8.874	0.598	4.633
RNN [45]	0.965	1.531	0.295	1.772	1.023	1.687	0.446	1.913	1.758	5.867	0.702	3.587
ARMA [11]	2.660	13.278	1.482	3.917	2.419	9.773	1.269	3.279	6.767	70.930	3.513	14.071
AR [11]	2.772	14.358	1.484	4.391	2.501	10.733	1.626	3.254	4.713	26.280	3.241	12.592

	qaq1				ramo				tixi			
	mae (TECU)	mse (TECU ²)	min (TECU)	max (TECU)	mae (TECU)	mse (TECU ²)	min (TECU)	max (TECU)	mae (TECU)	mse (TECU ²)	min (TECU)	max (TECU)
proposed	0.779	1.072	0.221	1.979	1.093	2.066	0.339	2.506	0.891	1.335	0.442	1.605
BILSTM [30]	1.123	1.839	0.317	2.200	1.393	3.317	0.358	3.889	0.912	1.431	0.494	2.049
LSTM [44], [31]	1.022	1.775	0.225	3.441	1.247	2.807	0.392	3.256	0.913	1.423	0.480	1.904
RNN [45]	0.889	1.213	0.235	1.496	1.461	3.414	0.541	4.266	1.141	2.043	0.423	2.673
ARMA [11]	2.942	13.571	2.078	4.070	5.914	56.487	4.032	7.588	1.919	7.242	1.392	2.877
AR [11]	2.995	14.612	1.883	4.100	5.671	56.737	3.195	8.384	2.150	8.641	1.021	2.637

B. Performance Evaluation and Comparison

In this section, we compare the various learning architectures to model the temporal dynamics of the ionospheric TEC. In particular, we implement and evaluate the traditional RNN, the unidirectional LSTM, the BILSTM, and the proposed CNN-GRU deep learning model (see Section III). We also compare our model performance against linear regressors of AR and ARMA. The comparison is carried out in terms of performance accuracy and computational efficiency.

Table III shows the performance evaluation metrics, as described in Section IV-B, of: 1) mae in TEC units (TECU); 2) mean squared error (mse) in TECU²; and 3) minimum (min) and maximum (max) TEC values again in TECU. The results in Table III have been calculated for the six stations (see Table II) and a randomly selected testing period of a whole day (October 9, 2018). In this table, we have also depicted the performance of the four compared nonlinear architectures (i.e., the proposed CNN-GRU, the BILSTM, the LSTM, and the RNN) along with the traditional AR and ARMA models. As is observed, the proposed CNN-GRU model has the best performance, in most cases, with a mae value between the interval [0.7 – 1.8]. On the other hand, the worst performance is for the AR and ARMA linear models with a mae value between [2.2 – 5.7]. Regarding the compared deep learning models, the mae values are as follows: RNN lies in [0.9 – 1.8], LSTM in [0.8 – 2.2], and BILSTM in [0.8 – 2.2]. The same conclusions are drawn for the other evaluation metrics.

Fig. 3 illustrates the Quantile–Quantile (Q–Q) plots for the “iqal” station between the ground truth and the evaluated deep learning architectures. In the case of a perfect performance, the points of the Q–Q plot will lie on the line $y = x$ (the red line in Fig. 3). The closer the data points are to the red line, the better is the model accuracy. Thus, the proposed CNN-GRU method has a better performance compared to the other ones since the blue points are closer to the red line. We also observe that higher vtec values are more challenging to be accurately modeled. However, again, the proposed CNN-GRU model achieves better performance for these high vtec values compared to the other models.

TABLE IV

50%, 65%, AND 98% PERCENTILE SCORES FOR THE EVALUATION OF THE DIFFERENT RNN-BASED NETWORKS WITH THE PROPOSED ONE

	percentile error (TECU)					
	graz			iqal		
	50%	65%	98%	50%	65%	98%
proposed	0.90	1.34	2.70	0.45	0.73	1.81
BILSTM	1.03	1.55	2.69	0.48	0.74	1.65
LSTM	1.12	1.45	2.91	0.44	0.73	1.82
RNN	1.00	1.55	2.86	0.59	0.95	2.61

	mal2			qaq1		
	50%	65%	98%	50%	65%	98%
	proposed	1.62	2.48	3.88	0.87	1.50
BILSTM	1.54	2.25	4.66	1.17	1.56	2.35
LSTM	1.93	2.52	4.69	1.00	1.43	2.50
RNN	1.21	1.66	4.38	0.85	1.28	3.09

	ramo			tixi		
	50%	65%	98%	50%	65%	98%
	proposed	0.87	1.26	3.04	0.67	0.97
BILSTM	1.09	1.67	3.56	0.77	1.13	2.47
LSTM	0.91	1.50	3.57	0.68	1.18	2.53
RNN	0.88	1.45	3.34	0.86	1.28	2.62

Table IV illustrates the 50%, 65%, and 98% percentile scores for the proposed and the compared nonlinear models. The proposed models of the “graz,” “iqal,” and “tixi” ground stations have the best performance, whereas the model of the “mal2” ground station presents the worst one. Again, the proposed CNN-GRU model has a better performance against the compared architectures. Based on the values of Table IV, we have concluded that the proposed CNN-GRU model, in the case of 98% percentile, achieves an improvement of 16.77% with respect to the RNN architecture and 12.71% and 9.49% with respect to the LSTM and BILSTM models, respectively. This means that the proposed CNN-GRU modeling error is not as large as in the case of the compared models.

Table V illustrates the average mae error in TECU over the six stations for the training and validation sets. The results have been obtained for all the nonlinear models. In most cases, the proposed CNN-GRU model has a better performance than the other methods. In particular, from Table V, CNN-GRU

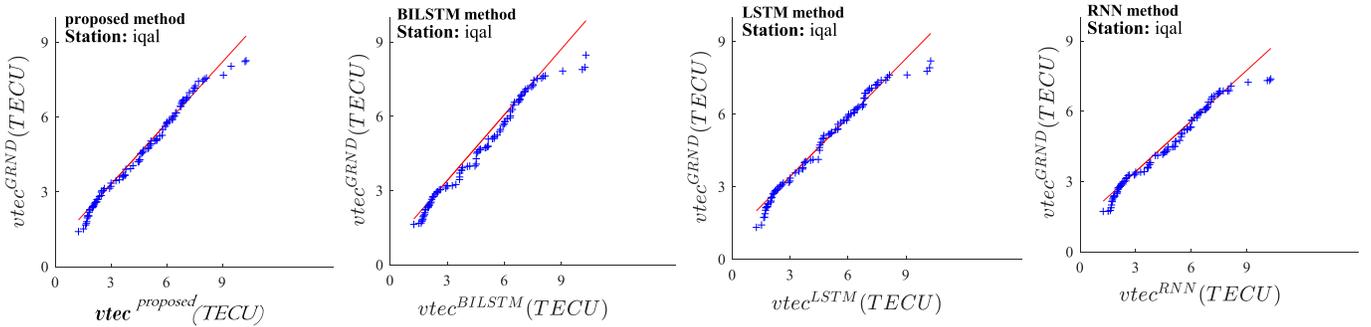


Fig. 3. Q-Q plots for the “iqal” station between the ground truth and the other deep learning techniques used for comparison.

TABLE V
PERFORMANCE METRIC MAE VALUES FOR THE TRAINING AND VALIDATION SETS AND THE SIX STATIONS AMONG THE RNN -BASED METHODS (GRU, BILSTM, LSTM, AND RNN)

	graz		iqal		mal2		qaq1		ramo		tixi	
	train	val	train	val.	train	val.	train	val	train	val.	train	val.
proposed	0.0167	0.0177	0.0206	0.0213	0.0164	0.0166	0.0239	0.0250	0.0115	0.0135	0.0298	0.0240
BILSTM	0.0168	0.0190	0.0213	0.0240	0.0185	0.0206	0.0252	0.0270	0.0115	0.0129	0.0299	0.0317
LSTM	0.0175	0.0189	0.0217	0.0243	0.0191	0.0201	0.0254	0.0268	0.0118	0.0135	0.0301	0.0303
RNN	0.0184	0.0214	0.0235	0.0295	0.0204	0.0241	0.0258	0.0281	0.0137	0.0157	0.0319	0.0364

achieves an average mae of 1.98% for the training set and 1.97% for the validation set. The other nonlinear models have an average mae of BILSTM 2.05% (training set) and 2.25% (validation set); LSTM 2.09% (training set) and 2.23% (validation set); and RNN 2.23% (training set) and 2.59% (validation set). This means that the proposed CNN-GRU model has an improvement of 11.07% and 23.9% with respect to the conventional RNN architecture in the training and validation sets, respectively. In addition, the proposed CNN-GRU model achieves an improvement of 3, 69% and 12.65% with respect to the BILSTM model for the training and validation sets. It should be mentioned that the improvement in the validation set is more significant than one of the training sets since the latter refers to data that they have not to be used during training.

The processing time in minutes and the number of the required trainable parameters per method are listed in Table VI. As is observed, the most efficient is the RNN architecture, which requires 23 min (for 200 training epochs), while the heaviest is the BILSTM model needing 280 min (for the same 200 epochs). The proposed CNN-GRU architecture requires 67 min for its training (again for 200 epochs).

The abovementioned processing time refers to the training phase of the models. It should be mentioned that training is carried out once. After model training is completed and the model parameters (weights) have been estimated, the time needed for modeling the TEC values is negligible. In particular, our trained CNN-GRU model requires 0.96 s to model the TEC values of over a period of a half year (test dataset: the second half of 2018).

Fig. 4 illustrates the training and validation learning curves, as computed for four evaluated models, for the “ramo” station. As is observed, all models converge to an acceptable mae error after 200 training epochs.

TABLE VI

COMPARISON AMONG DIFFERENT RNN-BASED METHODS ARCHITECTURE (GRU, BILSTM, LSTM, AND RNN) CHARACTERISTICS AND PARAMETERS (TRAINABLE PARAMETERS AND PROCESSING TIME)

	Methods			
	proposed	BILSTM	LSTM	RNN
Trainable Parameters (num.)	112493	379485	149453	38573
Epochs	200	200	200	200
Training Time (min)	67	280	104	23

C. Performance Evaluation for Stations at Different Latitudes and Days of Different Ionosphere Activities

Fig. 5 illustrates the VTEC values versus time for the four evaluated nonlinear models (CNN-GRU, LSTM, BILSTM, and RNN) at three different ground stations. In this figure, we also depict the ground-truth data as a dashed line. The satellites selected in this figure are GNSS visible from the sites: “graz” (mid-latitude), “mal2” (near the equator), and “tixi” (aurora region). Also, the satellites depicted in this figure have been selected in various time intervals during the day from each site, focusing more where the highest values of the ionospheric variability are observed. For each station/site, two days are presented in this figure: 1) September the 3rd, which is a day with low ionospheric activity and 2) September the 11th, in which high ionosphere activity is observed. For these two days, the values of Dst and F10.7 parameters are illustrated in the top-left diagram of Fig. 6. As observed from this figure, September 11 is a day of increased solar activity. On the contrary, September 3 is a day with normal solar and geomagnetic conditions. As expected, the model performance on 3/9 is slightly better in contrast to estimations during the of 11/9, in which abnormal ionosphere variability is observed [see Fig. 5]. However, the model catches the

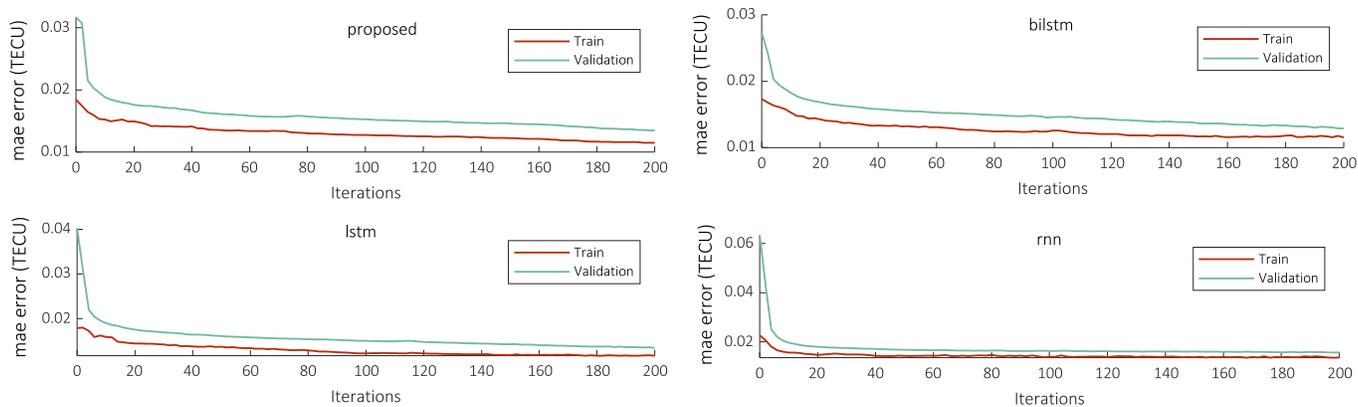


Fig. 4. Loss curve per RNN-based method (GRU, BILSTM, LSTM, and RNN) for training and validation sets for the ramo station.

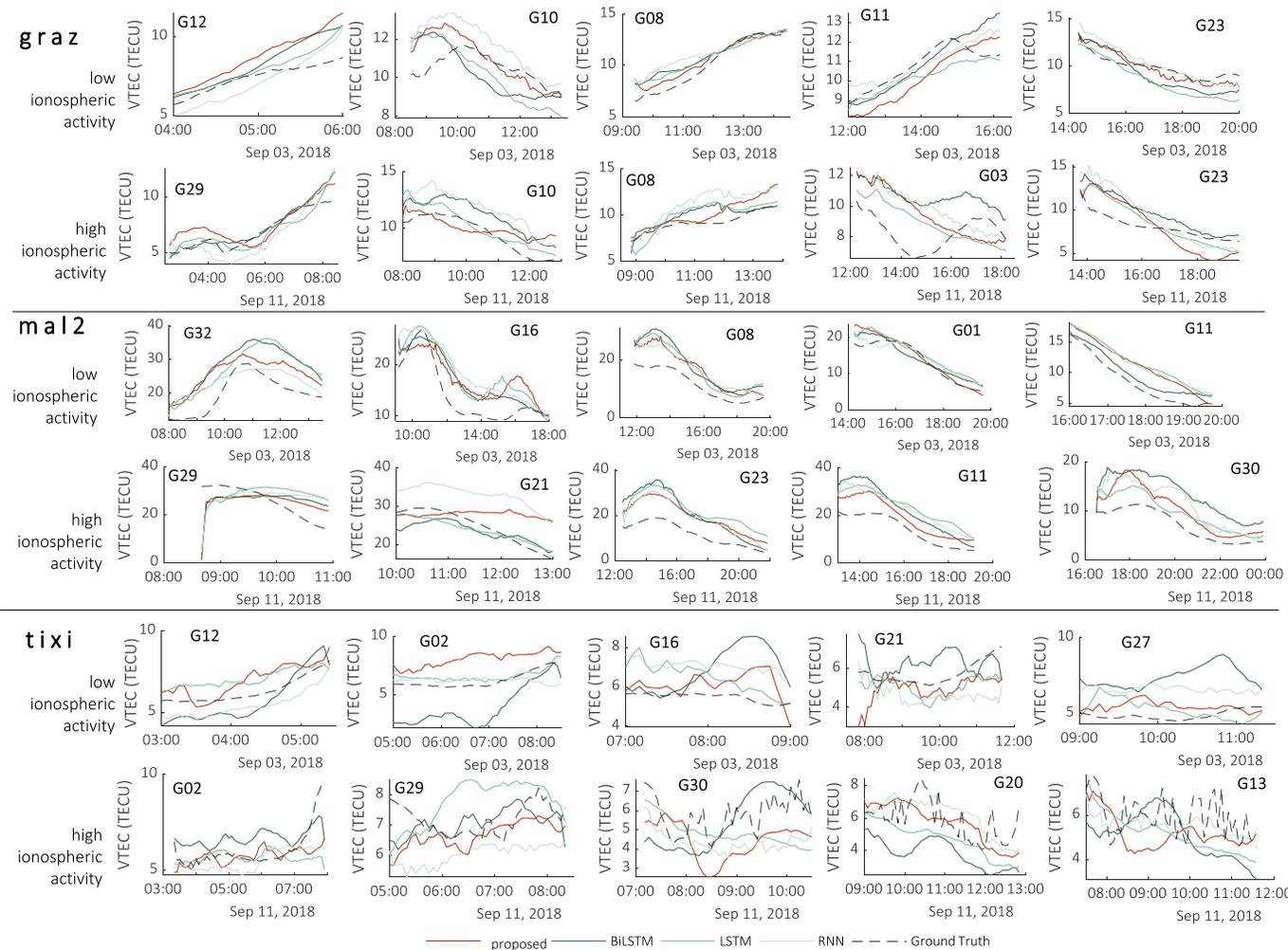


Fig. 5. VTEC for individual PRNs (unique pseudorandom noise that each GPS satellite transmits) between the GPS ground-truth values and GRU, BILSTM, LSTM, and RNN methods in various hours two different days: a day of low ionospheric activity (3/9) and a day of high ionospheric activity (11/9). VTEC values estimated per satellite from a low-latitude station (“mal2”) and a mid-latitude station (“graz”) and a high-latitude station (“tixi”).

trend of the line in both cases. Also, the model for the “graz” mid-latitude station shows better performance than the respective model of the “tixi” station. Overall, what is immediately noticeable in this figure is the CNN-GRU model’s ability to adequately estimate VTEC values for every satellite separately.

Fig. 6 shows the predicted VTEC values per station for a time period of a month (September 2018). The predicted

values of the proposed CNN-GRU model are illustrated with green color, whereas the red line is the ground-truth data. The top-left figure shows the F10.7 and DST values for the days of September. As observed, CNN-GRU predicted values achieve good performance. Also, the proposed CNN-GRU model catches the local maximum VTEC values during the days of increased ionospheric activity as mentioned above (i.e., September 11).

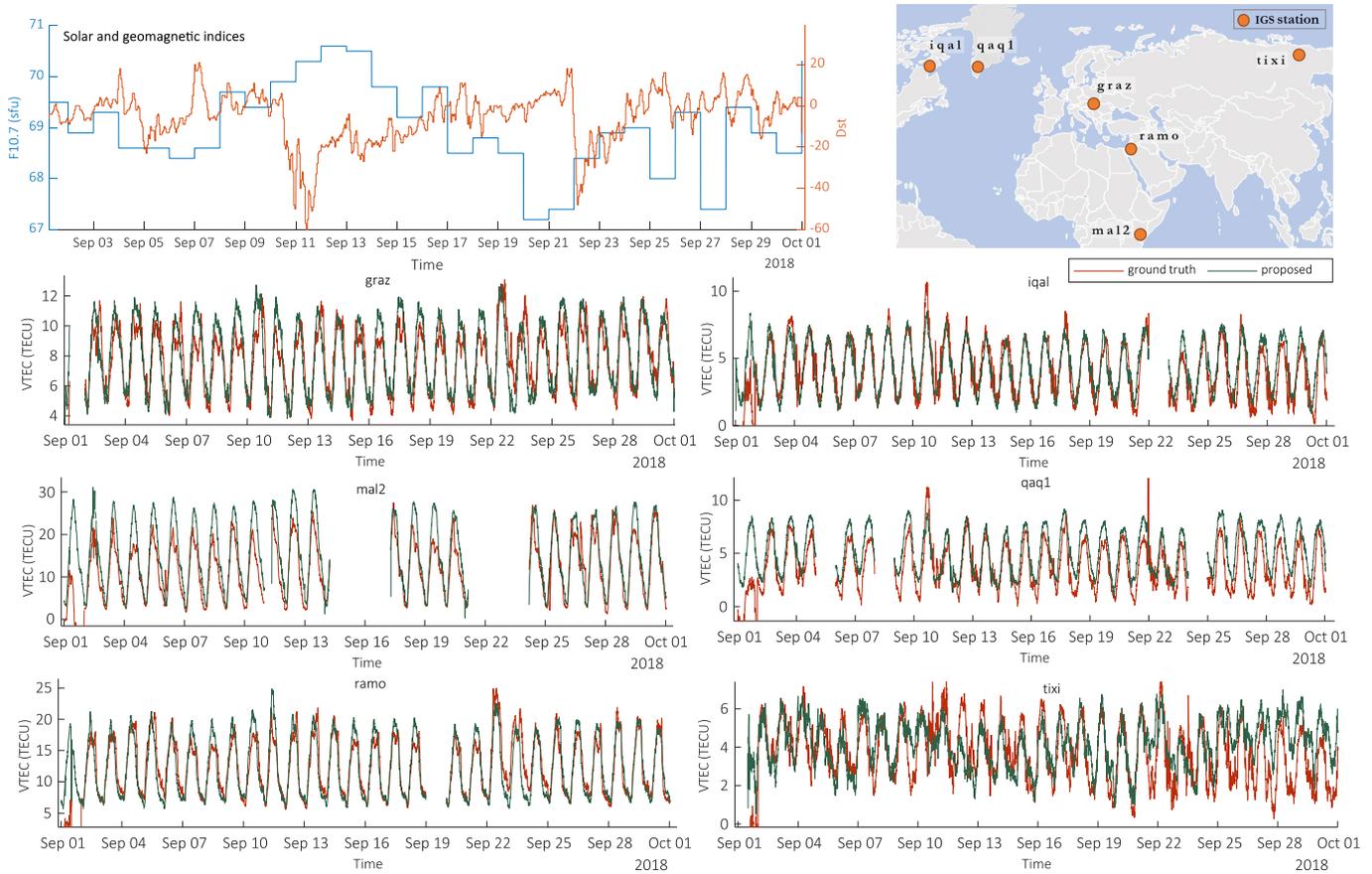


Fig. 6. (Top Left) Daily variations of the solar F10.7 index and the Dst-index. (Top Right) Map shows the location of each station. (Bottom) Diagrams illustrate the VTEC values per station on September 2018.

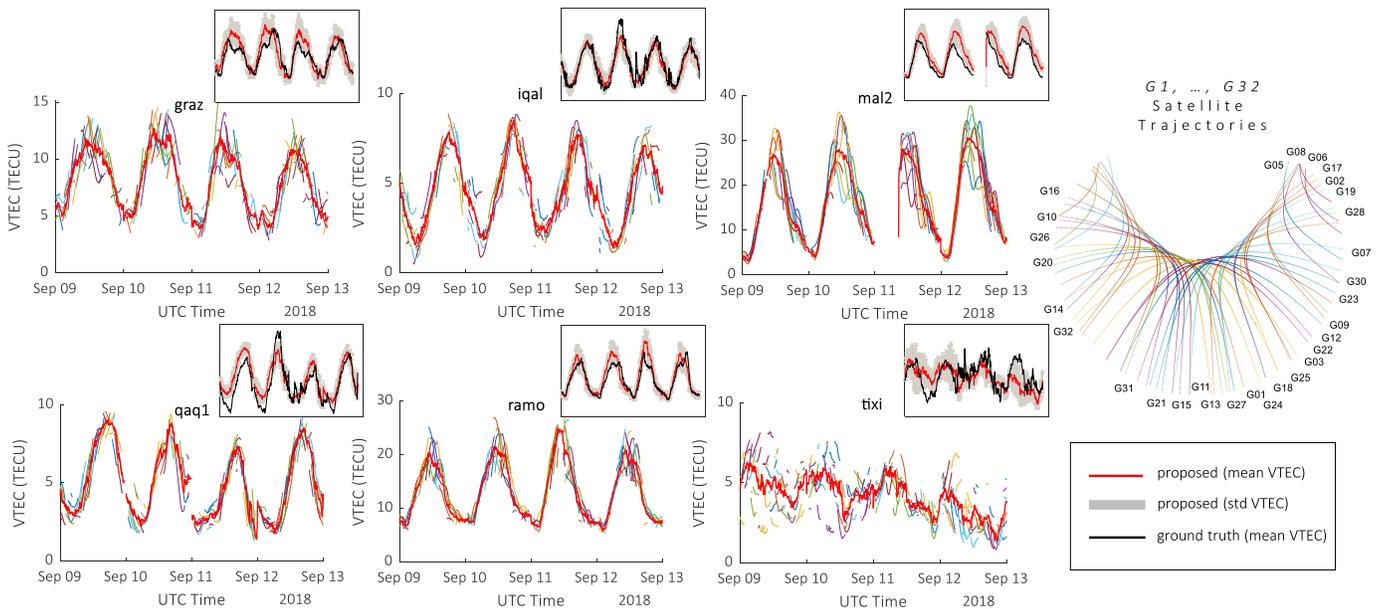


Fig. 7. VTEC sequences per satellite (colorful lines) of each ground station for GRU method for four days on September 2018 and the mean VTEC time series per station (red line). On top of each diagram, the mean red line per station is compared to mean ground-truth data per station, as illustrated with a black line.

Fig. 7 shows the estimated mean VTEC values, as illustrated with a red line, for each station, for four different days between September 9 and 13, 2018, compared to ground-truth values with a black line. Here, we can observe the diurnal changes, as the VTEC values are varying during each day, depending

on the electron density in the ionosphere. As observed, the ionospheric delay changes slowly through a daily cycle. It has its minimum values (2–4 TECU) between midnight and early morning and reaches its peak during the daylight hours. The “mal2” stations reach the bigger mean VTEC values

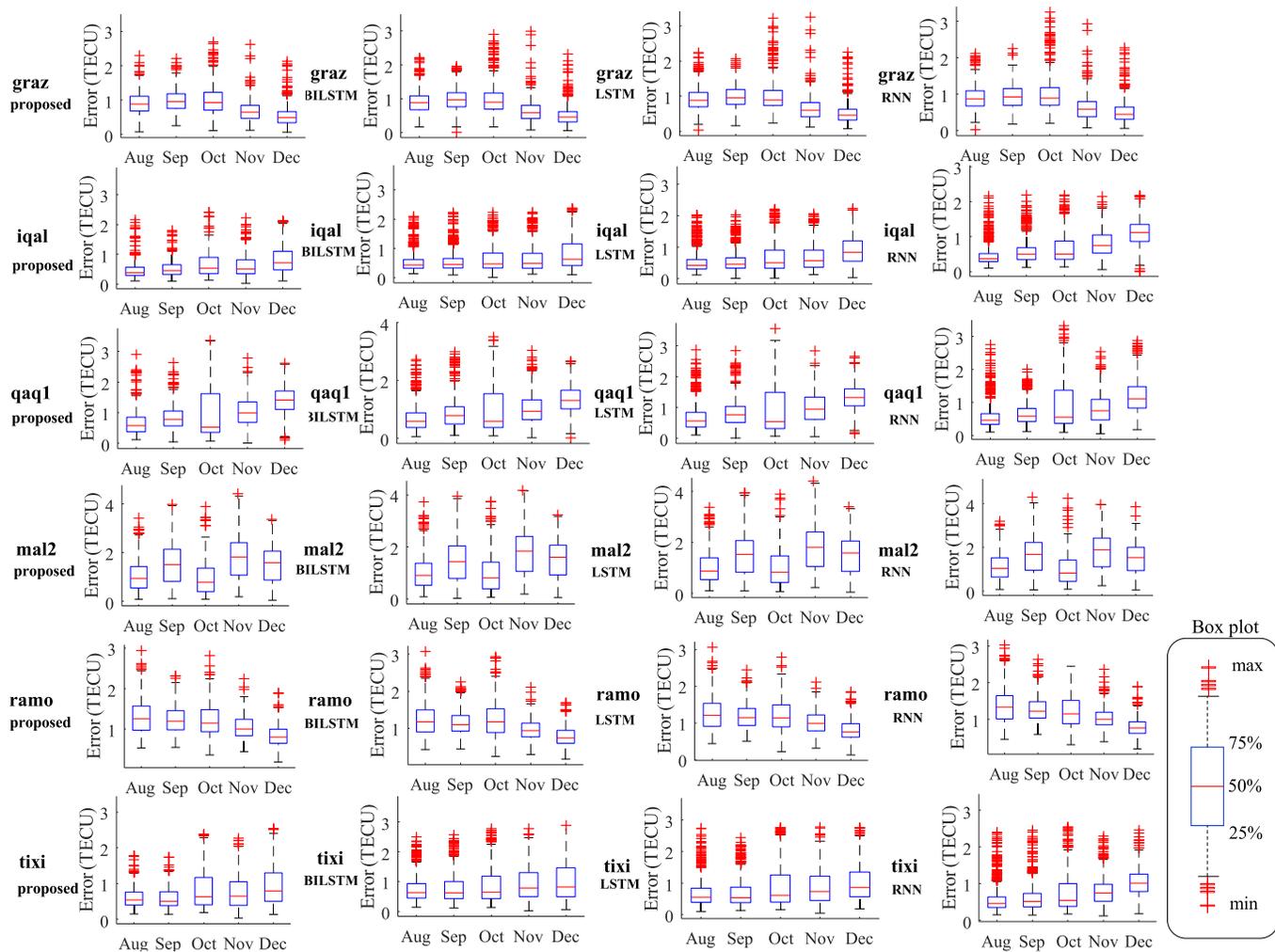


Fig. 8. Monthly box plots for the testing period during the second half of 2018 per station.

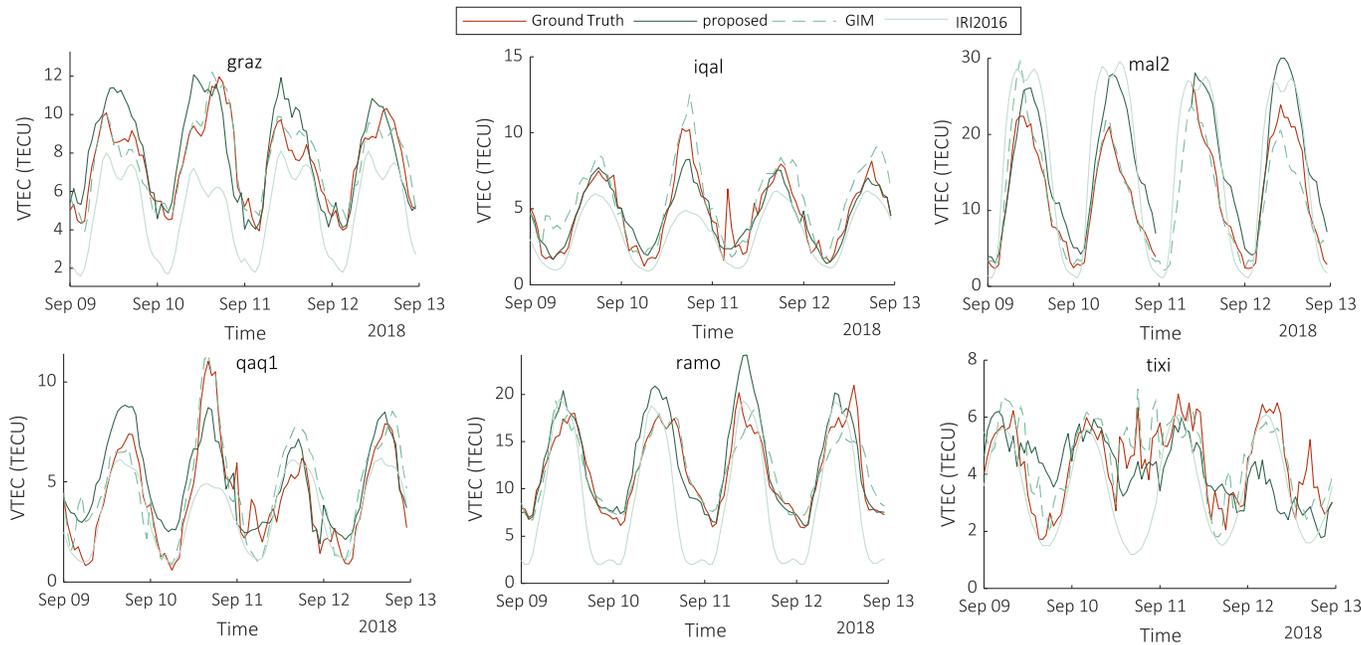


Fig. 9. Comparison between VTEC (TECU) RNN-based methods and the estimated GPS ionospheric VTEC values (ground truth), as well as the VTEC models (GIM, NeQuick, IRI2001, and IRI01-cor), for days between September 9 and 12.

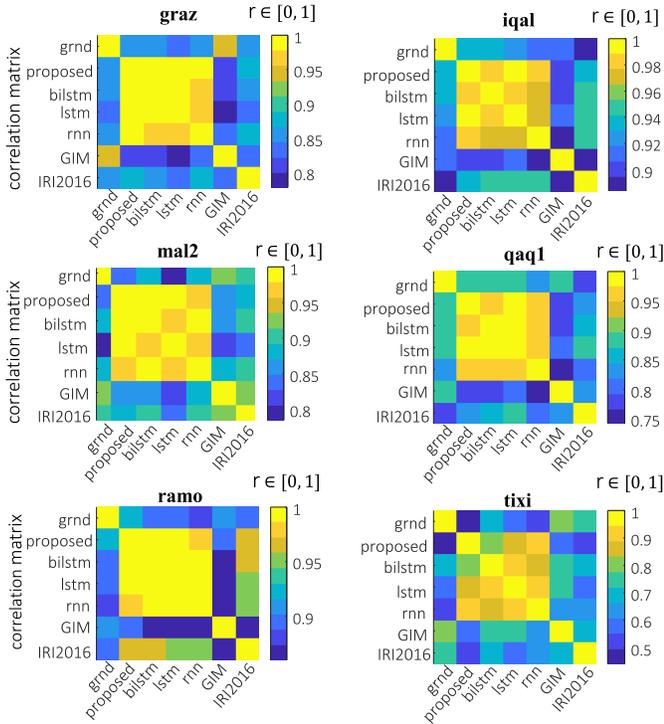


Fig. 10. Correlation matrices per station between VTEC (TECU) RNN-based estimations (GRU, BILSTM, LSTM, and RNN) and the estimated GPS ionospheric VTEC values (ground truth), as well as the VTEC models (GIM and IRI2016), for the time interval between September 9 and 12. In most of the stations, except for the “tixi” station in the aurora region, the correlation coefficient r ranges in the interval $[0.75, 1)$.

(~ 30 TECU), whereas the “tixi” station (located in the aurora region) has the minimum observed values. However, the “tixi” station appears with various fluctuations. The mean standard deviation values are 1.20 for “graz,” 0.80 for “iqal,” 2.81 for “mal2,” 0.70 for “qaq1,” 1.78 for “ramo,” and 1.04 for “tixi” ground station. This is illustrated in Fig. 7 where we have depicted with gray the standard deviation values.

D. Performance Evaluation for Different Months

At every time instance (point), the difference between the ground-truth VTEC values and the estimated ones from the various nonlinear learning models is computed. Thus, for the test time period of half a year (second half of 2018), we have an error time series for every station, deriving from the timewise differences of VTEC values (see Section IV-B). Fig. 8 depicts the distribution of these errors per month for the testing period (second half of the year 2018). Each box plot graphically depicts groups of vtec error values through their quartiles. Outliers are plotted as individual points, with the red cross symbol. The horizontal red line is the median value of the vtec error (50%), whereas the first and third quartiles, that is the 25th and 75th percentiles, respectively, are illustrated in a black dashed line. As observed, the performance evaluation per month is slightly different; however, these differences are not significant. In most of the cases, the proposed CNN-GRU method has a better performance as it concerns the outlier values, which are slightly fewer compared to other methods outliers (difference errors $> 75\%$). In August, there are 18% fewer outliers in the proposed CNN-GRU compared to the

RNN method. In the other months, this percentage is 5% for September, 13% for October, 17% for November, and 9% for December.

E. Performance Evaluation Between Different Ionosphere Models

Fig. 9 shows the mean VTEC values at every station, for IRI2016 and GIM TEC estimates compared to the ground-truth values and CNN-GRU TEC estimations, during four days between September 9 and 12. The VTEC maxima are shown for all stations between 8:00 and 12:00 A.M. In most cases, the CNN-GRU predicted values are similar to the ground truth. GIM-aided TEC values are also close to CNN-GRU TEC values. IRI2016 values seem to underestimate the ionospheric variability compared to the ground-truth TEC and TEC values from the GIM model.

Fig. 10 shows the correlation coefficient between the proposed CNN-GRU, the compared nonlinear learning methods, the ground-truth GPS TEC data, and the values derived from IRI2016 and GIM models. Yellow blocks indicate high correlation, whereas blue blocks show smaller values for the correlation index r . Overall, the stations “iqal,” “mal2,” and “ramo” attain higher correlation values ($r > 0.75$), whereas “tixi” (located near the equator) performs the worst values of correlation.

VI. CONCLUSION

In this article, we have proposed a combined CNN-GRU deep learning architecture for TEC modeling. The proposed model has been compared with other linear (e.g., AR and ARMA) and nonlinear (RNN, LSTM, and BILSTM) regression methods. Our main conclusions are given as follows.

- 1) The proposed CNN-GRU model achieves an improvement of 16.77% with respect to the RNN architecture and 12.71% and 9.49% with respect to the LSTM and BILSTM models, respectively, in the case of 98% percentile. This implies the capability of the proposed CNN-GRU network to capture TEC values both in normal and high solar activity periods.
- 2) The average mae improvement of the proposed model compared to the traditional RNN architecture is 23.9% with respect to the validation set, a set that it has not been used during training. In the case of the BILSTM model, the mae improvement is 12.65%.
- 3) The computational complexity of the proposed model is retained low although its performance accuracy is higher. In particular, as far as model testing is concerned, our CNN-GRU architecture requires less than 1 s (0.96 s) to model the TEC values over a period of a half year. Instead, regarding training, our model requires about 67 min to be trained for 200 epochs in contrast to the BILSTM architecture where training is executed within 280 min for the same number of epochs. It should be mentioned that the training of the model is carried out once.
- 4) The mean standard deviation value of mae of our model is 1.38 over all the six examined ground stations. The worst performance is for the “mal2” station, which is

located near the equator, while the best is for the “graz” station (mid-latitude).

As future work, we intend to assess the accuracy of the TECs computed by the model in the position domain, with precise coordinates of the IGS stations. Also, we will further investigate model reliability, given as inputs GNSS observational data from a regional network of stations close to each other, to examine the extent to which such models capture the regional anomalies. In addition, more complicated deep learning architectures, such as semisupervised learning [47], [48] and/or generative adversarial networks (GANs) [39], can be examined to see if they can improve the results while retaining small computational cost.

REFERENCES

- [1] A. Rovira-Garcia, J. M. Juan, J. Sanz, and G. González-Casado, “A worldwide ionospheric model for fast precise point positioning,” *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 8, pp. 4596–4604, Aug. 2015.
- [2] F. Meyer, R. Bamler, N. Jakowski, and T. Fritz, “The potential of low-frequency SAR systems for mapping ionospheric TEC distributions,” *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 4, pp. 560–564, Oct. 2006.
- [3] J. M. Forbes, S. E. Palo, and X. Zhang, “Variability of the ionosphere,” *J. Atmos. Sol.-Terr. Phys.*, vol. 62, no. 8, pp. 685–693, 2000.
- [4] H. Jiang *et al.*, “Assessment of spatial and temporal TEC variations derived from ionospheric models over the polar regions,” *J. Geodesy*, vol. 93, no. 4, pp. 455–471, Apr. 2019.
- [5] P. Yin, C. N. Mitchell, P. S. J. Spencer, and J. C. Foster, “Ionospheric electron concentration imaging using GPS over the USA during the storm of July 2000,” *Geophys. Res. Lett.*, vol. 31, no. 12, pp. 1–4, Jun. 2004.
- [6] J. Klobuchar, “Ionospheric time-delay algorithm for single-frequency GPS users,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-23, no. 3, pp. 325–331, May 1987.
- [7] M. Hernández-Pajares *et al.*, “The ionosphere: Effects, GPS modeling and the benefits for space geodetic techniques,” *J. Geodesy*, vol. 85, no. 12, pp. 887–907, Dec. 2011.
- [8] A. Wang, J. Chen, Y. Zhang, L. Meng, B. Wang, and J. Wang, “Evaluating the impact of CNES real-time ionospheric products on multi-GNSS single-frequency positioning using the IGS real-time service,” *Adv. Space Res.*, vol. 66, no. 11, pp. 2516–2527, Dec. 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0273117720306396>
- [9] L. Wang, Z. Li, M. Ge, F. Neitzel, Z. Wang, and H. Yuan, “Validation and assessment of multi-GNSS real-time precise point positioning in simulated kinematic mode using IGS real-time service,” *Remote Sens.*, vol. 10, no. 2, p. 337, Feb. 2018. [Online]. Available: <https://www.mdpi.com/2072-4292/10/2/337>
- [10] Y. Kong, H. Chai, J. Li, Z. Pan, and Y. Chong, “A modified forecast method of ionosphere VTEC series based on ARMA model,” in *Proc. Forum Cooperat. Positioning Service (CPGPS)*, May 2017, pp. 90–95.
- [11] C. Wang, S. Xin, X. Liu, C. Shi, and L. Fan, “Prediction of global ionospheric VTEC maps using an adaptive autoregressive model,” *Earth, Planets Space*, vol. 70, no. 1, pp. 1–14, Dec. 2018.
- [12] G. K. Rangarajan and T. Iyemori, “Time variations of geomagnetic activity indices Kp and Ap: An update,” *Annales Geophysicae*, vol. 15, no. 10, pp. 1271–1290, Oct. 1997, doi: [10.1007/s00585-997-1271-z](https://doi.org/10.1007/s00585-997-1271-z).
- [13] I. Kutiev *et al.*, “Solar activity impact on the Earth’s upper atmosphere,” *J. Space Weather Space Clim.*, vol. 3, p. A06, 2013.
- [14] O. V. Mandrikova, N. V. Fetisova, R. T. Al-Kasasbeh, D. M. Klienskiy, V. V. Geppener, and M. Y. Ilyash, “Ionospheric parameter modelling and anomaly discovery by combining the wavelet transform with autoregressive models,” *Ann. Geophys.*, vol. 58, no. 5, p. 0550, Nov. 2015.
- [15] P. Karthik, D. V. Ratnam, N. Vanga, P. S. Brahmanadam, B. S. S. Rao, and K. S. Kumar, “Auto regressive ionospheric prediction model for GPS applications,” *Int. J. Comput. Appl.*, vol. 48, no. 4, pp. 7–9, Jun. 2012.
- [16] D. V. Ratnam, Y. Otsuka, G. Sivavaraprasad, and J. R. K. K. Dabbakuti, “Development of multivariate ionospheric TEC forecasting algorithm using linear time series model and ARMA over low-latitude GNSS station,” *Adv. Space Res.*, vol. 63, no. 9, pp. 2848–2856, May 2019.
- [17] O. V. Mandrikova, N. V. Glushkova, and I. V. Zhivet’ev, “Modeling and analysis of ionospheric parameters by a combination of wavelet transform and autoregression models,” *Geomagnetism Aeronomy*, vol. 54, no. 5, pp. 593–600, Sep. 2014.
- [18] A. D. Doulamis, N. D. Doulamis, and S. D. Kollias, “On-line retrainable neural networks: Improving the performance of neural networks in image analysis problems,” *IEEE Trans. Neural Netw.*, vol. 11, no. 1, pp. 137–155, Jan. 2000.
- [19] J. B. Habarulema, L.-A. McKinnell, P. J. Cilliers, and B. D. L. Opperman, “Application of neural networks to South African GPS TEC modelling,” *Adv. Space Res.*, vol. 43, no. 11, pp. 1711–1720, Jun. 2009.
- [20] F. Sabzehee, S. Farzaneh, M. A. Sharifi, and M. Akhoondzadeh, “TEC regional modeling and prediction using ANN method and single frequency receiver over IRAN,” *Ann. Geophys.*, vol. 61, no. 1, p. 103, Mar. 2018.
- [21] A. Yilmaz, K. E. Akdogan, and M. Gurun, “Regional TEC mapping using neural networks,” *Radio Sci.*, vol. 44, no. 3, pp. 1–16, Jun. 2009.
- [22] N. Bakalos *et al.*, “Protecting water infrastructure from cyber and physical threats: Using multimodal data fusion and adaptive deep learning to monitor critical systems,” *IEEE Signal Process. Mag.*, vol. 36, no. 2, pp. 36–48, Mar. 2019.
- [23] A. D. Doulamis, N. D. Doulamis, and S. D. Kollias, “An adaptable neural-network model for recursive nonlinear traffic prediction and modeling of MPEG video sources,” *IEEE Trans. Neural Netw.*, vol. 14, no. 1, pp. 150–166, Jan. 2003.
- [24] Z. Huang and H. Yuan, “Ionospheric single-station TEC short-term forecast using RBF neural network,” *Radio Sci.*, vol. 49, no. 4, pp. 283–292, Apr. 2014.
- [25] Z. Zhang, S. Pan, C. Gao, T. Zhao, and W. Gao, “Support vector machine for regional ionospheric delay modeling,” *Sensors*, vol. 19, no. 13, p. 2947, Jul. 2019.
- [26] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, “Deep learning for computer vision: A brief review,” *Comput. Intell. Neurosci.*, vol. 2018, pp. 1–13, Feb. 2018.
- [27] W. Zaremba, I. Sutskever, and O. Vinyals, “Recurrent neural network regularization,” 2014, *arXiv:1409.2329*. [Online]. Available: <http://arxiv.org/abs/1409.2329>
- [28] S. Hochreiter, “The vanishing gradient problem during learning recurrent neural nets and problem solutions,” *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 6, no. 2, pp. 107–116, Apr. 1998.
- [29] F. A. Gers, D. Eck, and J. Schmidhuber, “Applying LSTM to time series predictable through time-window approaches,” in *Artificial Neural Networks—ICANN*, G. Dorffner, H. Bischof, and K. Hornik, Eds. Berlin, Germany: Springer, 2001, pp. 669–676.
- [30] W. Sun, L. Xu, X. Huang, W. Zhang, T. Yuan, and Y. Yan, “Bidirectional LSTM for ionospheric vertical total electron content (TEC) forecasting,” in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2017, pp. 1–4.
- [31] M. Kaselimi, A. Voulodimos, N. Doulamis, A. Doulamis, and D. Delikaraoglou, “A causal long short-term memory sequence to sequence model for TEC prediction using GNSS observations,” *Remote Sens.*, vol. 12, no. 9, p. 1354, Apr. 2020.
- [32] A. Ruwali, A. J. S. Kumar, K. B. Prakash, G. Sivavaraprasad, and D. V. Ratnam, “Implementation of hybrid deep learning model (LSTM-CNN) for ionospheric TEC forecasting using GPS data,” *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 6, pp. 1004–1008, Jun. 2021.
- [33] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” 2014, *arXiv:1409.3215*. [Online]. Available: <http://arxiv.org/abs/1409.3215>
- [34] M. Kaselimi, N. Doulamis, A. Voulodimos, E. Protopapadakis, and A. Doulamis, “Context aware energy disaggregation using adaptive bidirectional LSTM models,” *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3054–3067, Jul. 2020.
- [35] M. Kaselimi, N. Doulamis, A. Doulamis, and D. Delikaraoglou, “A sequence-to-sequence temporal convolutional neural network for ionosphere prediction using GNSS observations,” *Int. Arch. Photogram., Remote Sens. Spatial Inf. Sci.*, vol. 43, pp. 813–820, Aug. 2020.
- [36] F. Zhou, D. Dong, W. Li, X. Jiang, J. Wickert, and H. Schuh, “GAMP: An open-source software of multi-GNSS precise point positioning using undifferenced and uncombined observations,” *GPS Solutions*, vol. 22, no. 2, pp. 1–10, Apr. 2018.
- [37] N. Wang, Y. Yuan, Z. Li, O. Montenbruck, and B. Tan, “Determination of differential code biases with multi-GNSS observations,” *J. Geodesy*, vol. 90, no. 3, pp. 209–228, Mar. 2016.

- [38] N. Bergeot *et al.*, "Near real-time ionospheric monitoring over Europe at the Royal Observatory of Belgium using GNSS data," *J. Space Weather Space Clim.*, vol. 4, p. A31, 2014.
- [39] M. Kaselimi, N. Doulamis, A. Voulodimos, A. Doulamis, and E. Protopapadakis, "EnerGAN++: A generative adversarial gated recurrent network for robust energy disaggregation," *IEEE Open J. Signal Process.*, vol. 2, pp. 1–16, 2021.
- [40] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," 2013, *arXiv:1312.6026*. [Online]. Available: <http://arxiv.org/abs/1312.6026>
- [41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [42] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000.
- [43] A. Graves, "Generating sequences with recurrent neural networks," 2014, *arXiv:1308.0850*. [Online]. Available: <https://arxiv.org/abs/1308.0850>
- [44] I. Srivani, G. S. V. Prasad, and D. V. Ratnam, "A deep learning-based approach to forecast ionospheric delays for GPS signals," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 8, pp. 1180–1184, Aug. 2019.
- [45] R. Tang, F. Zeng, Z. Chen, J.-S. Wang, C.-M. Huang, and Z. Wu, "The comparison of predicting storm-time ionospheric TEC by three methods: ARIMA, LSTM, and Seq2Seq," *Atmosphere*, vol. 11, no. 4, p. 316, Mar. 2020.
- [46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [47] N. Doulamis and A. Doulamis, "Semi-supervised deep learning for object tracking and classification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 848–852.
- [48] N. Doulamis and A. Voulodimos, "FAST-MDL: Fast adaptive supervised training of multi-layered deep learning models for consistent object tracking and classification," in *Proc. IEEE Int. Conf. Imag. Syst. Techn. (IST)*, Oct. 2016, pp. 318–323.



Maria Kaselimi received the Diploma degree and the M.Sc. degree in geoinformatics from the National Technical University of Athens (NTUA), Athens, Greece, in 2015 and 2017, respectively, where she is pursuing the Ph.D. degree.

She has 20 articles in international journals and conferences and more than 85 citations. She has been involved in several European research projects as a researcher. Her research interest focuses on machine learning, signal processing techniques, and data analysis and modeling with applications in the

fields of energy, geosciences, and the environment.



Athanasios Voulodimos (Member, IEEE) received the Dipl.Ing., M.Sc., and Ph.D. degrees (Hons.) from the School of Electrical and Computer Engineering, National Technical University of Athens (NTUA), Athens, Greece, in 2007, 2010, and 2011, respectively.

He is an Assistant Professor with the Department of Informatics and Computer Engineering, University of West Attica, Psachna, Greece. He has been involved in several European research projects as a senior researcher and a technical manager. He has

coauthored more than 130 articles in international journals, conference proceedings, and books in the research areas of machine learning and signal processing. He has received more than 2400 citations.

Dr. Voulodimos received awards for his academic performance and scientific achievements.



Nikolaos Doulamis received the Diploma and Ph.D. degrees (Hons.) in electrical and computer engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 1995 and 2001, respectively.

He is an Associate Professor with NTUA. He is the author of more than 75 (240) journals (conference) articles in the field of signal processing and machine learning. He received more than 5450 citations. He is involved in large-scale European projects, such as H2020 Stop-It and H2020 Benefice.

Dr. Doulamis received many awards (e.g., the Best Student Among all Engineers and the Best Paper Awards). He has served as an organizer and/or a TPC in major IEEE conferences.



Anastasios Doulamis (Member, IEEE) received the Diploma and Ph.D. degrees in electrical and computer engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 1995 and 2001, respectively.

Until January 2014, he was an Associate Professor with the Technical University of Crete, Chania, Greece. He is an Assistant Professor with NTUA. He is the author of more than 350 articles in leading journals and conferences receiving more than 5400 citations.

Dr. Doulamis received several awards in his studies, including the Best Greek Student Engineer and the Best Graduate Thesis Award. He has also served on the program committee of several major conferences of IEEE and ACM.



Demitris Delikaraoglou received the Diploma degree in surveying engineering from the National Technical University of Athens, Athens, Greece, in 1974, the M.Sc. degree from the Wolfson College, University of Oxford, Oxford, U.K., in 1976, and the Ph.D. degree from the University of New Brunswick, Fredericton, NB, Canada, in 1980.

He was an Associate Professor of geodesy with the National Technical University of Athens from 2009 to 2020. He has more than 1000 citations and more than 50 publications in journals and conferences.

Dr. Delikaraoglou is also a member of the American Geophysical Union (AGU) and a reviewer of various scientific journals.