IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 April 2023, accepted 26 April 2023, date of publication 1 May 2023, date of current version 6 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3272057

RESEARCH ARTICLE

Novel Deep Hybrid and Ensemble Algorithms for Improving GPS Navigation Positioning Accuracy

TOLGA AYDIN AND EBRU ERDEM
Department of Computer Engineering, Ataturk University, Yakutiye, 25240 Erzurum, Turkey
Corresponding author: Tolga Aydin (atolga@atauni.edu.tr)

ABSTRACT GPS (Global Positioning System) has been a widespread system used for various purposes in today's world and it is essential to suggest innovative effective solutions to improve its use and functions. The present study proposes GPS coordinate conversion models based on Machine Learning (ML) and Deep Learning (DL) algorithms in order to "improve accuracy of GPS conversion and positioning services". 23 different models are tested on two different data sets to achieve this purpose. The study primarily aims to improve positioning accuracy of navigation systems by using GPS data through hybrid and ensemble algorithms. The proposed DL-based models are named as GPSCNNs and GPSLSTM. GPSCNNs contain "Xception, VGG16, VGG19, Alexnet, CNN1, CNN2, CNN3" deep learning algorithms in their structure. Of these algorithms, "Xception, VGG16, VGG19, Alexnet" are pre-trained models. "CNN1" consists of 2 Convolution, 2 Average Pool, 1 Flatten, and 5 Dense layers. "CNN2" consists of 1 Convolution, 1 Max Pool, 1 Flatten, and 4 Dense layers. "CNN3" consists of 4 Convolution, 4 Batch Normalization, 2 Max Pool, 1 Flatten, and 3 Dense layers. GPSLSTM contains 1 LSTM and 1 Dense layer in its structure. Raw GPS data are fed into the models as input, which was followed by obtaining information about the features of the data and getting coordinate data as input. The results show that ensemble models provide the most accurate positioning and GPSCNNs and GPSLSTM were quite promising in boosting this accuracy.

INDEX TERMS Ensemble algorithms, GPS, GPSCNNs, GPSLSTM.

I. INTRODUCTION

There are various technologies utilizing mobile communication tools to offer services to locate outer and inner spaces. Among these technologies are Wi-fi finger print, wireless sensor network, and wireless local area network (WLAN). Positioning-based mobile applications use embedded GPS technology to locate a place or building etc. in the world. Positioning-based systems using GPS data can be employed for different purposes, such as vehicle/aircraft tracking, drive assistance, shipping by fleets, data/information sharing, civil applications such as aviation, security, telecommunication, atmosphere-related measurements and analyses, and navigation systems. GPS data provide position information when it receives direct signals from four or more satellites [1]. Delays in the propagation of signals are measured to locate a specific position.

The associate editor coordinating the review of this manuscript and approving it for publication was Giacomo Fiumara.

There are two angular values providing information about GPS coordinates or position of a location in the world: latitude and longitude. GPS can calculate latitude, longitude, altitude and time zone values of a location by processing satellite signals [2]. Latitude refers to the angular distance of a place in reference to the equator. If this value is between 0^0 and 90^0, this place is in the northern hemisphere, and if it is between 0^0 and -90^0, the place is located in the southern hemisphere. Longitude is an angular coordinate of a point defining its place in reference to the Greenwich Meridian (the prime meridian). While a place is in the east of Greenwich if a longitude value between 0^0 and 180^0, the place is in the west of the prime meridian if the value is between 0^0 and -180^0.

Different methods might be adopted to minimize positioning error by using GPS data. These include Assisted GPS (AGPS), Differential GPS (DGPS), Wide Area Augmentation System (WAAS), and some additional peripheral modules and software algorithms, such as Kalman filter and Wiener filter. The study by Krasuski et al. [3] implements a mathematical algorithm by using DGPS technique to increase aircraft positioning accuracy. The results show that this algorithm significantly improves the positioning accuracy. However, some of the modules proposed to minimize positioning error are costly and some of the recursive predictor filters lack self-correction ability [2], [4]. Artificial intelligence based studies were also conducted to overcome these limitations and improve positioning accuracy. Bhatt et al. propose a new information-based source difference artificial neural network for advanced positioning. The difference of this model from the classical multi-layer perceiver (MLP) is that the model consists of two MLP neural networks: difference model and coarse model. The proposed model proves to be effective for GPS/Inertia Navigation System integration schemes [5]. GPS signals might become unstable due to satellite clock error, signal jamming, and some effects of GPS receivers and external factors in the nearby environment, which in turn might lead to GPS data interruptions. Significantly disrupting navigation system accuracy, GPS data interruption problem is especially observed in urban areas, which are often characterized by closed areas, tall buildings and tunnels. Among the techniques recommended to solve this problem are sensors and AI methods [6]. The study by Li et al. introduces a fuzzy neural network based GPS/INS odometer system and shows that this system might improve position and speed accuracy in case of long lasting GPS interruptions [7]. Another study aiming to increase positioning accuracy during GPS interruptions combines Strong Tracking Kalman Filter (STKF) and Wavelet Neural Network (WNN) algorithms. This new model is implemented to examine the experimental results on GPS and INS data collected in a land vehicle test. The results show that it is possible to make highly precise corrections for INS during GPS interruptions [8]. The study by El-Shafie et al. seeking to minimize positioning error during GPS interruptions and improve the performance of vehicle navigation system proposed a model based on Wavelet-Adaptive Neural Fuzzy Inference System (ANFIS). They aim at predicting the location of the vehicle in a reliable way by obtaining successful results from Neuro-fuzzy module and road test [9]. Noureldin et al. propose a multi-sensor integration approach to combine the data obtained from GPS and INS. Such a system uses Wavelet Multiple Resolution Analysis (WMRA) and Artificial Neural Network (ANN). WMRA module is utilized to compare INS/GPS position outputs while ANN module is employed to locate the vehicle accurately and to predict positioning errors. Their findings reveal that Neuro-wavelet technique is successful in positioning accuracy [1]. Kaygısız et al. integrate GPS/INS system and ANN to improve positioning accuracy during GPS interruptions and system positioning error was reduced to less than 9% [10]. The study by Zhang et al. suggests the "Multiple-Decrease Factor Cubature Kalman Filter (MDF-CKF) and a hybrid algorithm based on random forest (MDF-CKF/RF). They test the effectiveness of MDF-CKF/RF) by using the data obtained from off-road vehicles' navigation practices. They obtain a better improvement in positioning accuracy compared to MDF-CKF/RF and CKF [11]. Zhang et al. employ "Kalman Filter-Gradient Boosting Decision Tree-Particle Swarm Optimization (KGP)" technique to improve INS/GPS navigation positioning accuracy. Their experimental results indicate that this technique significantly improves the positioning accuracy relative to MLP and random forest regression (RFR) [12]. Xiong et al. develop "Square-Root Cubature Kalman Filter (SRCKF)" and a dual-model system based on RFR. They attain a remarkable improvement in navigation accuracy thanks to the proposed model [13].

It is possible to determine the factors that affect positioning accuracy by using different coordinate conversion algorithms. To illustrate, Wang et al. [14] propose a GPS coordinate conversion framework based on K-Medoid clustering algorithm. As proven by an example practice, the proposed framework is suitable for implementation. Similarly, Song et al. develop a GPS coordinate conversion method based on neural network. The experimental practices obtain promising statistical findings, which implies high levels of applicability for the method [15]. The coordinate conversion algorithms are given more emphasize now due to increasing demand for positioning accuracy in different fields, such as technology and engineering. Thus, the contributions of the present study to the related fields can be listed as follows:

- It introduces an innovative positioning approach providing predictions through proposed models by using a large amount of data involving coordinate position values obtained in the past such as latitude, longitude etc.
- In order to improve positioning accuracy, in addition to other models recommended in the literature, the study implements deep hybrid neural network and ensemble learning models in addition to other models recommended in the literature.
- The study found that ensemble and hybrid models are superior to single models (Xception, Vgg16, Vgg19, AlexNet, LSTM, CNN).
- As far as we know, the proposed hybrid and ensemble models have been implemented on two different anonymous data set for the first time in the literature.

II. MATERIAL AND METHODOLOGY

A. MATERIAL

The first data set [16] (dataset#1) includes data about housing estates in California. Table 1 introduces the following information about these estates: latitude, longitude, housing median age, total rooms, total bedrooms, population, households, median income and median house value. This data set includes 17000 pieces of data. GPS data in the second data set [17] (dataset#2) belong to various mosquito species (Culex Erraticus, Culex Pipiens, Culex Restuans, Culex Pipiens/Restuans, Culex Salinarius, Culex Tarsalis, Culex Territans) in Chicago, a city located in Illinois in the USA. West Nile Virus (Wnv) is a disease spread by mosquitos [18]. In the data set, a "Wnv present" value of 1 indicates the presence of the disease while the value 0 means the absence of the disease.

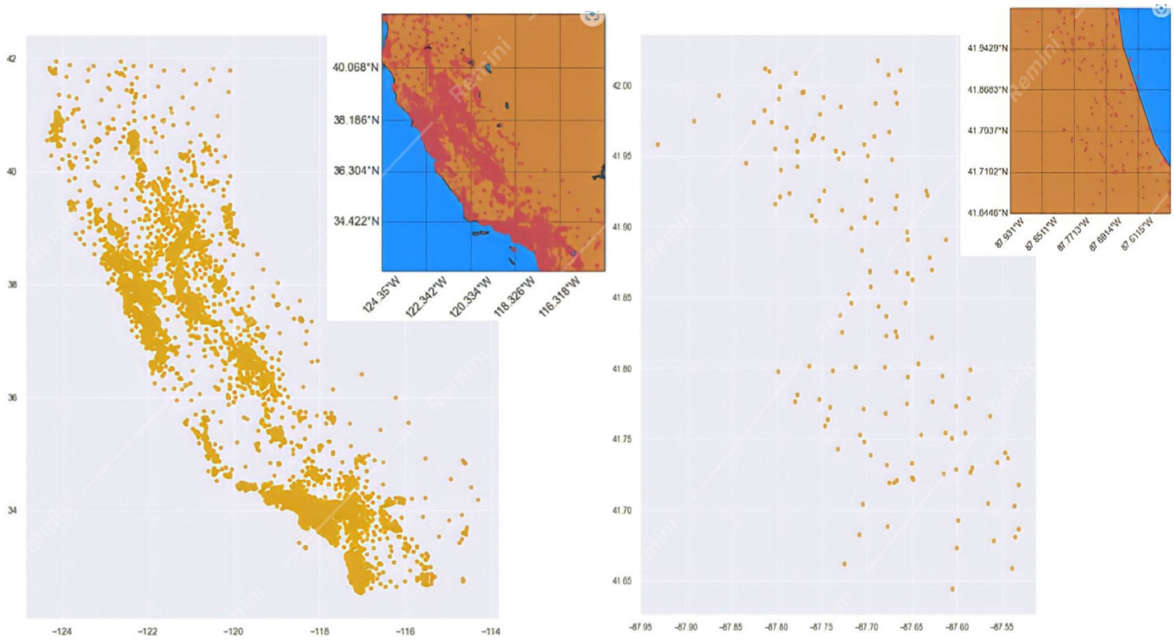

FIGURE 1. Plotting coordinates and mapping of geographic points of dataset#1 and dataset#2, respectively.

TABLE 1. Detailed information available in the data sets.

Data Set	Features	Data Type
dataset#1	Longitude	decimal
dataset#1	Latitude	decimal
dataset#1	Housing Median Age	float
dataset#1	Total Rooms	float
dataset#1	Total Bedrooms	float
dataset#1	Population	float
dataset#1	Households	float
dataset#1	Median Income	float
dataset#1	Median House Value	float
dataset#2	Date	date
dataset#2	Address	string
dataset#2	Species	string
dataset#2	Block	int
dataset#2	Street	string
dataset#2	Trap	string
dataset#2	Address Number and Street	string
dataset#2	Latitude	decimal
dataset#2	Longitude	decimal
dataset#2	Address Accuracy	int
dataset#2	Number Mosquitos	int
dataset#2	Wnv Present	int

TABLE 2. CNN1, CNN2, CNN3, and LSTM parameter details.

	CNN1 Architecture	CNN2 Architecture	CNN3 Architecture	LSTM Archite cture
Number of Layers	10	7	14	2
Number of Layers Layers	10 (Conv2D, AveragePool2 D, Conv2D, AveragePool2 D, Flatten, Dense, Dense, Dense, Dense, Dense)	7 (Conv2D, MaxPool2D, Flatten, Dense, Dense, Dense, Dense)	14 (Conv2D, BatchNorm alization, Conv2D, BatchNorm alization, MaxPool2D , Conv2D, BatchNorm alization, Conv2D, BatchNorm alization, MaxPool2D , Flatten, Dense, Dense, Dense)	2 (LSTM , Dense)
Units	(relu, -, relu, - , -, relu, relu, relu, relu, linear)	(relu, -, -, relu, relu, relu, linear)	(-, -, -,-,-,-,- ,-,-,-, relu, relu, linear)	(tanh, linear)
Number of Units	(256, -, 256, - , -, 256, 128, 64, 40, 1)	(16, -, -, 512, 256, 128, 1)	(32, -, 32, -, -, 64, -, -, -, 4096, 4096, 1)	(256, 1)
Dropout	(0.3, 0.3, 0.3, 0.3, 0.3)	(0.1)	(0.5)	(-)
Learning Rate/Optimizer	0.001/Adam	0.001/Adam	0.001/Adam	0.001/ Adam

The data set also includes the following information: number of mosquitos in each address; latitude and longitude values of different addresses in the north, south, west and east of the city; and speed trap numbers of the addresses (T002, T007, T129, T148, T235, etc.). Trap numbers can be considered a reference name, which is useful in regulating travelers' speed by warning them against traffic surveillance cameras and speed tracking devices. Another data available in the data set is "Address Accuracy", which is a Canadian mail program designed in order to improve delivery services by ensuring accurate locationing [19]. Address information in this data set can be referred to as "information to locate an animal". The data set comprises the data collected at different times between 2007 and 2013. Table 1 below presents the related details. This data set consists of 10506 pieces of data. Fig. 1

TABLE 3. Meta model parameter details.

Model	Parameters
SVR	kernel="rbf", gamma=0.12, C=1.7, epsilon=0.11, verbose=1

tabulates information about the coordinates and geographical locations in dataset#1 and dataset#2.

B. METHODOLOGY

The process involving the selection and regulation of a model is called optimization. Through optimization, the internal structure, loss functions, and coefficients that make up the models might be altered. The goal of this approach is not simply to find a solution but also to seek a solution that optimizes the performance criteria and minimizes the loss criteria, such as prediction error. However, these models cannot always guarantee the best solution. This study proposes ensemble and hybrid models to improve the model performance. As far as we know, there are not any studies focusing on improving positioning accuracy by using GPS data through hybrid and ensemble algorithm models. The present study is expected to contribute to the literature by providing invaluable findings regarding this issue. The overall framework of the models is presented in Fig. 2.

1) MACHINE LEARNING BASED ENSEMBLE MODELS

Combining predictions from different models might provide better results than using predictions from the best model among different models. In the present study, the predictions from different regression models are combined. The method combining predictions obtained from different models is called ensemble method. Heterogen ML models are combined by employing ensemble methods such as voting and stacking. In the first model architecture, voting among k nearest neighbor (kNN) and linear regression models are employed, while in the second model architecture, voting among k nearest neighbor (kNN) and decision tree regression (DTR) models are utilized. The most basic ensemble model is "voting" method. The means of predictions from different models are calculated in order to apply voting to regression models. In the third model architecture, "stacking" is employed. While in the voting method the weight of each model is the same, in the stacking method the weight of each of the N models is learned ($w_1, w_2, \ldots, w_g$), (N:3), so that the predicted value ($\hat{y}$) is close to real value (y).

Stacking uses a meta-learning algorithm to learn how best to combine predictions from multiple base machine learning algorithms. The stacking model architecture consists of Level 1 and Level 2. Level 1 involves more than one base model and the models fit on the training data whose predictions are compiled. Level 2 involves a meta model combining predictions from base models and learns how to combine base models as effectively as possible. The meta model is trained for predictions by base models via out-of-sample data. In the present study, outcomes of base models used as input for meta model are real values in case of regression. The predictions made by the meta model are the combination of predictions made by the base models or a weighted mean of these predictions. The advantage of stacking is that it can make use of the abilities of models displaying good performances in a regression problem and make better predictions than any single model in the ensemble. For a given ensemble stacking model, let us assume that the initial training data set is represented by "V" consisting of "g" observations and "f" features. In this situation, some base models are trained via V data set. "P_i ($i \in 1, 2, \ldots, g$)" prediction results coming from each model become the input of the second level (meta-model). The essence of stacking is to stack the predictions obtained by linear combination ("$y_1, y_2, \ldots, y_g$") and the weights ($w_1, w_2, \ldots, w_g$) in weight vector (w) proposed by meta-learner.

$$\hat{y}_{stacking}(x) = \sum_{n=1}^{g} (w_n y_n(x)) = w_1\hat{y}_1 + w_2\hat{y}_2 + \ldots + w_g\hat{y}_g \quad (1)$$

It is already known how to fit the model into the formula given in (1) and "y" is predicted through linear regression. However, the predictions by each model in the formula ($\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_N$) build up the features. Base models were trained on the training data set in the stacking ensemble model using 5-fold cross validation technique. Fig. 3 displays the overall diagram that belongs to the stacking model. The proposed architecture employs kNN and DTR, support vector regression (SVR) as the base model and LR as the meta model. First of all, the obtained predictions are stacked in a two-column series in order to learn weights from kNN, DTR, and SVR models. Later, LR model is trained on these predictions. The prediction made by meta model is formed by combining predictions made by base models.

In the present study, the first model is called "kNN+LR", the second one "kNN+DTR" and the third one "ensemble stacking".

2) DEEP LEARNING BASED HYBRID MODELS (GPSCNNS AND GPSLSTM)

"7 different architectures were designed for deep hybrid models. The proposed deep hybrid models are called GPSCNNs and GPSLSTM. GPSCNNs consists of CNN based models, which are Xception+SVR, VGG16+SVR, VGG19+SVR, Alexnet+SVR, CNN1+SVR, CNN2+SVR, CNN3+SVR. The model forming GPSLSTM is LSTM based and it is called LSM+SVR. While the base models forming the hybrid model are Xception, VGG16, VGG19, Alexnet, CNN1, CNN2, CNN3 DL networks or LSTM network, the meta model consists of SVR. Fig. 4 presents the diagram for DL based hybrid models.

All pre-trained models (Xception, VGG16, VGG19, AlexNet) are CNN-based modes. In ILSVRS 2014 competition, VGGNet is trained on ImageNet data set. VGGNet [20] consists of convolution, pooling, flattening and dense layers.

TABLE 4. Test metric performance results of models.

Dataset	Method	MAE	MSE	Dataset	Method	MAE	MSE
dataset#1	kNN+LR	0.4879	**0.4070**	dataset#2	kNN+LR	0.0671	0.0083
dataset#1	kNN+DTR	**0.4622**	0.4331	dataset#2	kNN+DTR	0.0504	0.0054
dataset#1	ensemble stacking	0.5783	0.7055	dataset#2	ensemble stacking	**0.0278**	**0.0033**
dataset#1	CNN1+SVR	1.6874	4.7705	dataset#2	CNN1+SVR	0.0783	0.0094
dataset#1	CNN2+SVR	1.2739	3.2715	dataset#2	CNN2+SVR	0.0796	0.0086
dataset#1	CNN3+SVR	1.6906	4.1305	dataset#2	CNN3+SVR	0.0745	0.0087
dataset#1	VGG16+SVR	1.7440	5.0164	dataset#2	VGG16+SVR	0.0783	0.0095
dataset#1	VGG19+SVR	1.7440	5.0164	dataset#2	VGG19+SVR	0.0783	0.0095
dataset#1	Xception + SVR	1.7438	5.0138	dataset#2	Xception+SVR	0.0783	0.0095
dataset#1	AlexNet + SVR	1.7161	4.8155	dataset#2	AlexNet+SVR	0.0783	0.0095
dataset#1	LSTM+SVR	1.7413	4.9396	dataset#2	LSTM+SVR	0.0786	0.0094
dataset#1	CNN1	43.3226	1880.0005	dataset#2	CNN1	29.9817	898.9148
dataset#1	CNN2	7.6211	86.4677	dataset#2	CNN2	3.6143	13.1092
dataset#1	CNN3	1.9665	4.9173	dataset#2	CNN3	1.4379	2.1475
dataset#1	SVR	1.7917	4.0250	dataset#2	SVR	0.6984	0.4973
dataset#1	VGG16	1.7035	4.6219	dataset#2	VGG16	0.1225	0.0185
dataset#1	VGG19	1.6713	3.9741	dataset#2	VGG19	2.9033	8.4386
dataset#1	Xception	1.9207	4.8862	dataset#2	Xception	0.2982	0.1289
dataset#1	AlexNet	39.1792	1729.2031	dataset#2	AlexNet	24.9750	732.1809
dataset#1	LSTM	1.8201	3.9905	dataset#2	LSTM	0.1040	0.0194
dataset#1	kNN	1.8152	4.2508	dataset#2	kNN	0.0364	0.0040
dataset#1	LR	1.8236	3.9803	dataset#2	LR	0.0778	0.0094
dataset#1	DTR	2.1729	7.8415	dataset#2	DTR	0.6984	0.4973

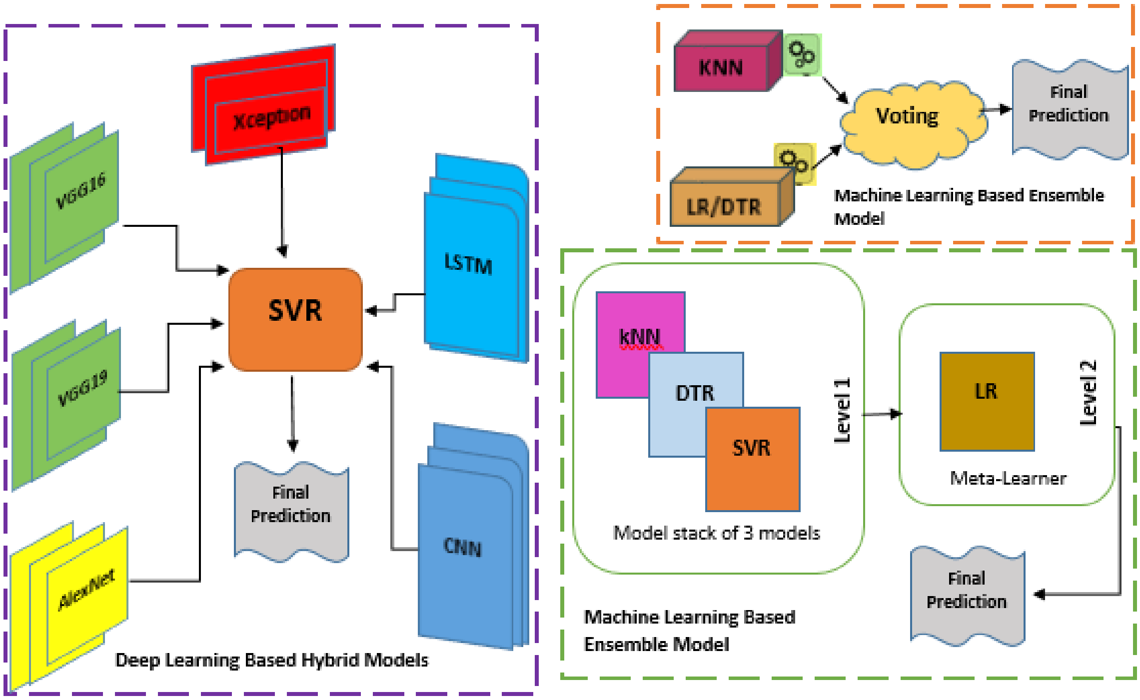

FIGURE 2. General framework of the proposed models.

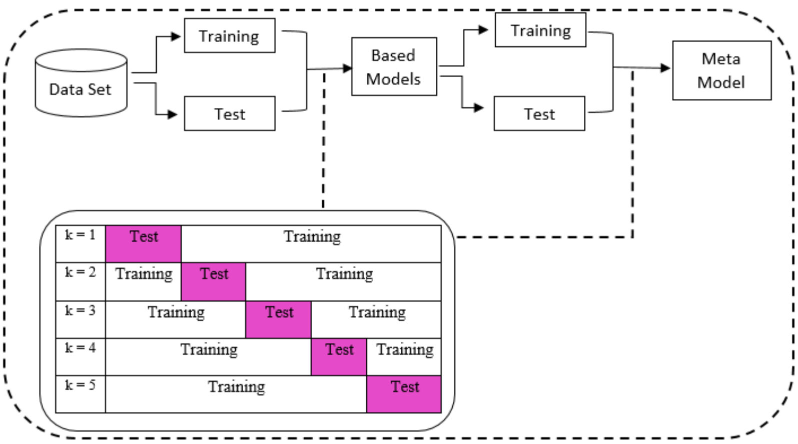

FIGURE 3. Diagram of stacking model.

While convolution layer consists of 3 × 3 filters and pooling layer is made up of 2 × 2 filters. The step size in pooling layer is 2. There are two pre-trained model structures that consist of 16 and 19 layers [21]. Two deep neural networks are used in the present study. Xception network was introduced by Chollet in 2016 [22]. This architecture employs depthwise separable convolution. CNN features are extracted in each model. AlexNet [23] is a large deep neural network that consists of 60 million parameters involving 3 convolution layers and 2 fully connected layers.

LSTM [24] operates on ordinal data and is a derivation of RNN. It is a model developed to eliminate the disadvantages of RNN. LSTM transition equation is given in (2-9). In this equation, i_t: input gate, f_t: forget gate, o_t: output gate, u_t: activation function, c_t: memory unit, h_t: hidden state output gate, w: weight matrices, b: ias vectors, $\sigma(\cdot)$: sigmoid activation function and tanh$(\cdot)$: tangent hyperbolic activation function. Weight matrices for the input layer x_t are w_{xi}, w_{xf} and w_{xu}. Weight matrices for the hidden layer output h_t are U_{hi}, U_{hf}, and U_{hu}. LSTM consists of input layer, parallel-connected memory blocks, and output layer. Input layer determines information updated by memory unit and a decision is made regarding which new information to store in the system (2). Forget gate determines information used for memory unit in the previous moment. This gate produces f_t between 0 and 1 according to current x_t and previous output h_{t-1}. The purpose of this process is to make a decision whether to allow c_{t-1} information produced in the

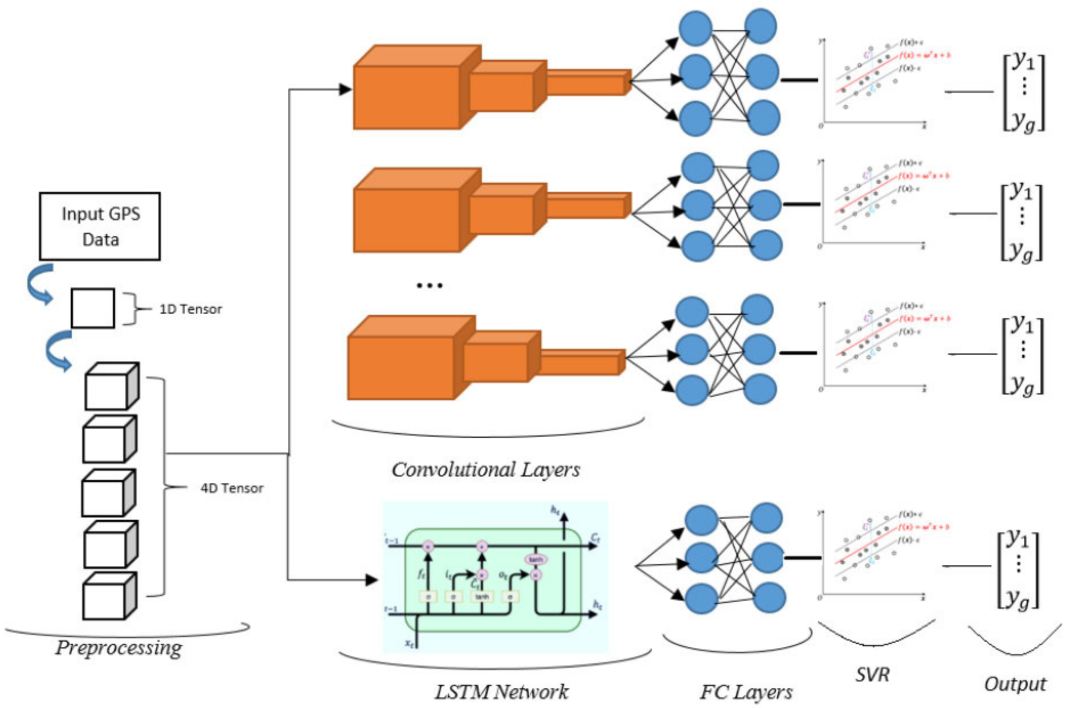

FIGURE 4. Diagram of DL based hybrid models.

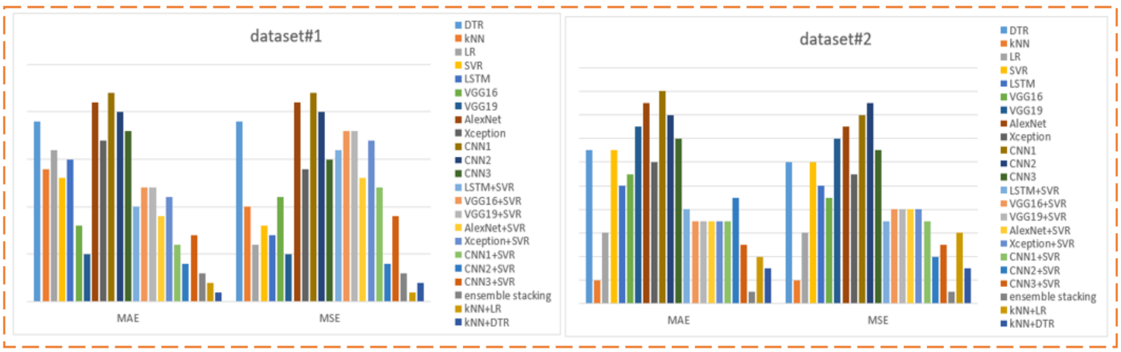

FIGURE 5. Tabulated summary of models' performances.

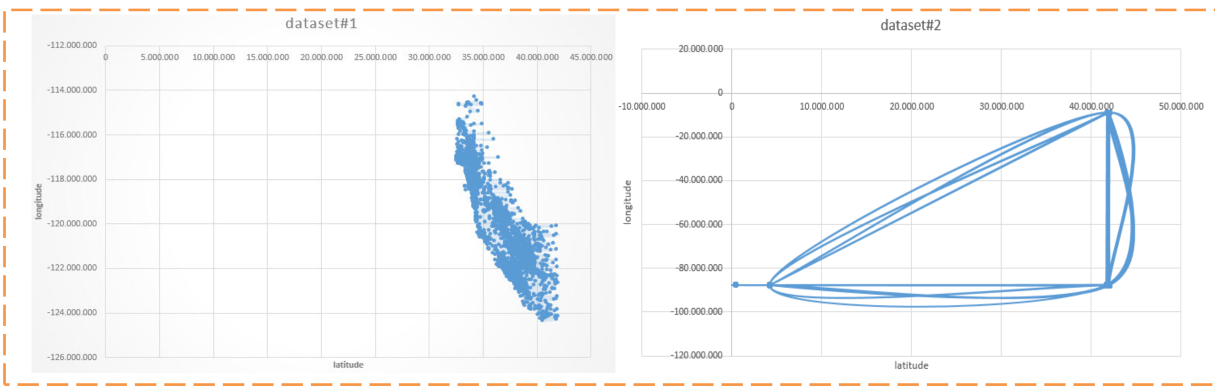

FIGURE 6. Graphic distributions of real longitude values (y) in the data set.

previous step to pass completely or partially (3). Output gate determines information output to hidden layer. Memory cell forms a vector consisting of c_t values that can be added to cell status. Eventually, a decision is made regarding which information to output (6). Later, c_t is set between -1 and +1 by using "tanh" activation. The obtained output is multiplied

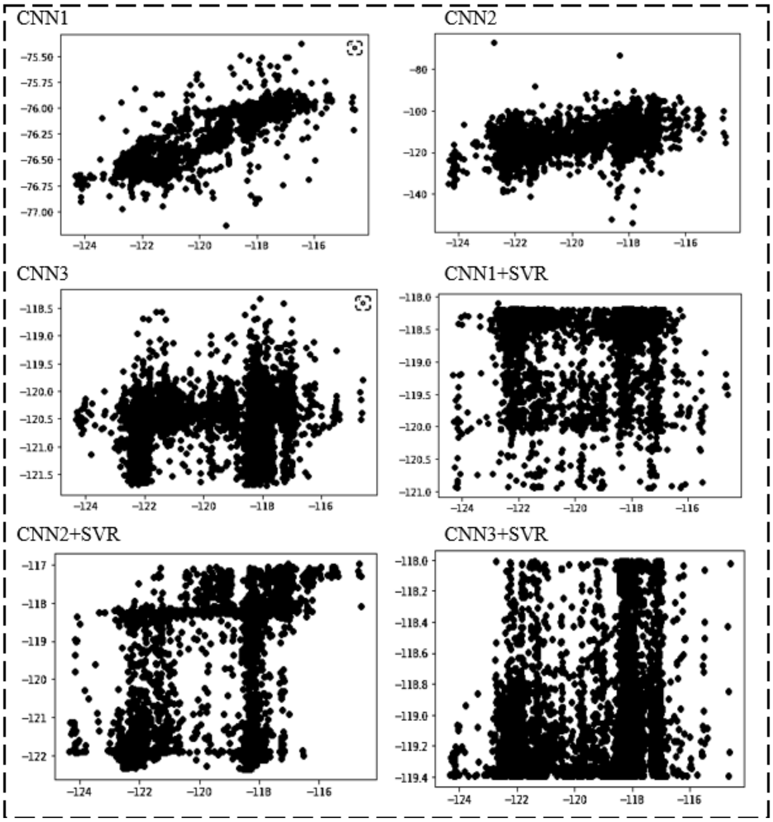

FIGURE 7. Prediction values by CNN models and GPSCNNs for dataset.

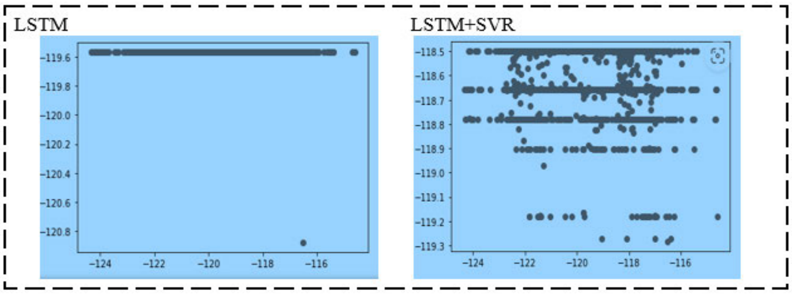

FIGURE 8. Prediction values by GPSLSTM for dataset#1.

by the value obtained by performing "sigmoid" activation, and target output "h_t" is obtained (7). The present study proposes a single-layer LSTM network.

$$i_t = \sigma\left(w_{xi}x_t + U_{hi}h_{t-1} + b_i\right) \tag{2}$$

$$f_t = \sigma\left(w_{xf}x_t + U_{hf}h_{t-1} + b_f\right) \tag{3}$$

$$u_t = tanh\left(w_{xu}x_t + U_{hu}h_{t-1} + b_u\right) \tag{4}$$

$$c_t = \left(i_t.u_t + f_t.c_{t-1}\right) \tag{5}$$

$$o_t = \sigma\left(w_{xo}x_t + U_{ho}h_{t-1} + b_o\right) \tag{6}$$

$$h_t = \left(o_t.\tanh\left(c_t\right)\right) \tag{7}$$

$$\sigma\left(x\right) = \frac{1}{\left(1+e^{-x}\right)} \tag{8}$$

$$tanh = \frac{e^x - e^{-x}}{\left(e^x + e^{-x}\right)} \tag{9}$$

One-dimensional GPS data are converted into 4-dimensional tensor and fed into convolutional based models as input. CNN based DL algorithms utilized for coordinate conversion of GPS data consist of two parts: forward propagation and backward propagation. First, (x, y) data are selected and x is taken as input value. Next, feature information passes through each layer and reaches output layer, and output value (Output_s) is calculated by utilizing "f" sigmoid activation function. Equation 10 formulates this process.

$$\text{Output}_s = f_n(f_{n-1}(f_{n-2}(\ldots(f_3\left(f_2\left(f_1\left(x_s w^1\right)w^2\right)w^3\right) \ldots)w^{n-2})w^{n-1})w^n) \tag{10}$$

In backward propagation, the difference between real and estimated value of y_s is calculated. Later, "Adam", a Gradient

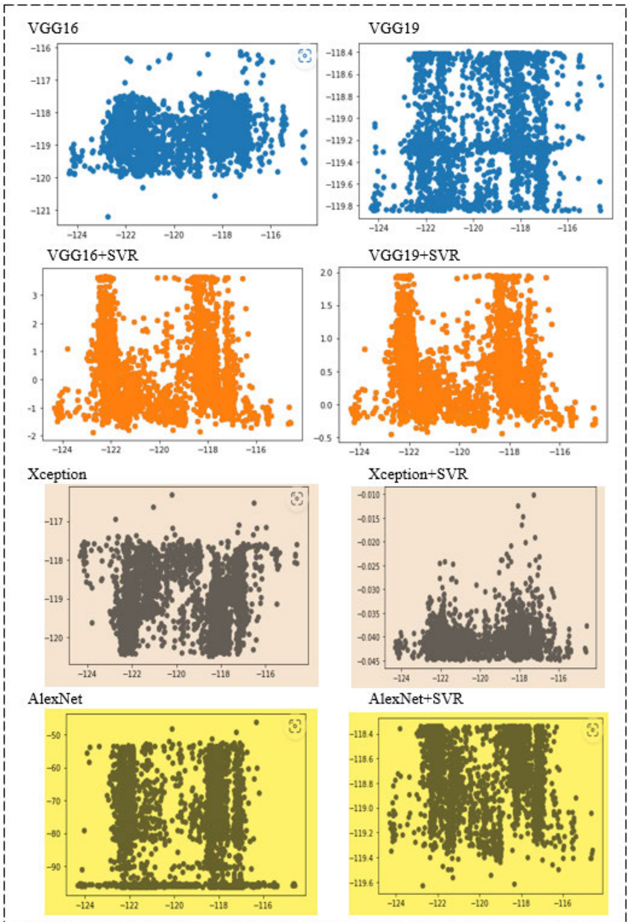

FIGURE 9. Prediction values by GPSCNNs and pre-trained models for dataset#1.

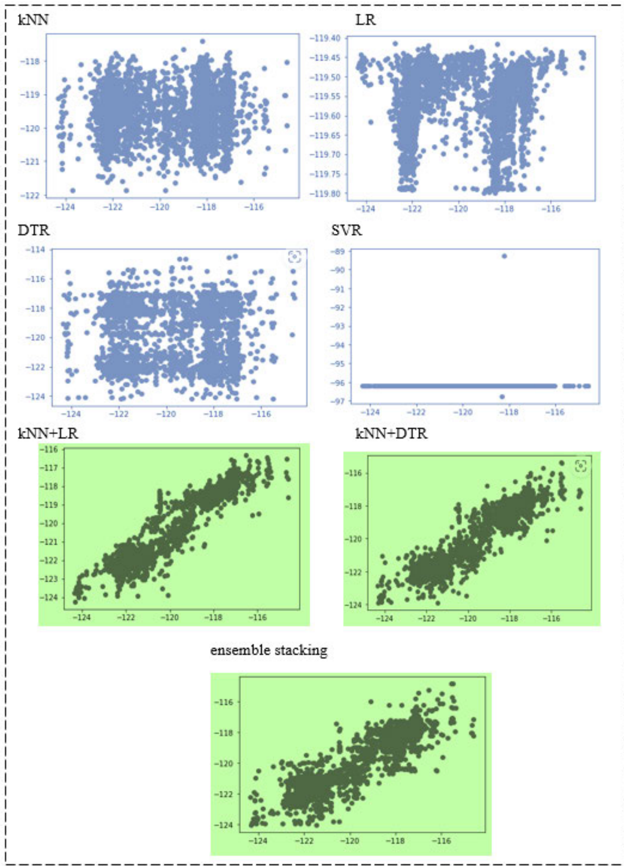

FIGURE 10. Prediction values by ML and ensemble models for dataset#1.

Descent Optimization algorithm, is employed to minimize errors and regulate weight matrix. In CNN based training, let I be the number of units in input layer, O be the number unit in output layer, and M be the number of units in hidden layer. Also let $\hat{w}_{ik}$: i represent weight of output unit to hidden layer and $\hat{w}_{kj}$: k weight of hidden layer to output layer. Input, output, and hidden layer output vectors are given in (11-13) respectively.

$$x = [x_0, x_1, \ldots, x_{\mathrm{I}-1}, x_{\mathrm{I}}](\text{Input vector}) \tag{11}$$

$$y = [y_0, y_1, \ldots, y_{\mathrm{O}-1}, y_{\mathrm{O}}](\text{Output vector}) \tag{12}$$

$$m = [m_0, m_1, \ldots, m_{\mathrm{M}-1}, m_{\mathrm{M}}](\text{Hidden layer output vector}) \tag{13}$$

Let ϕ_k represent hidden unit threshold value and ϕ_j the threshold value of output unit. Equations 15 and 16 gives output vector of each k hidden unit in hidden layer m_k and output vector of each j output unit in output layer y_{j}.

$$f(x) = \frac{1}{(1+e^{-jx})} \tag{14}$$

$$m_{\mathrm{k}} = f(\sum_{i=0}^{I-1} \hat{w}_{ik} x_i + \phi_k) \tag{15}$$

$$y_{\mathrm{j}} = (\sum_{i=0}^{M-1} w_j m_j + \phi_j) \tag{16}$$

The output obtained from CNN based base model is fed into SVR meta model as input. SVR is the regression version used for regression problems of Support Vector Machines [25] in prediction and regression problems.

Let D be the data set where $D = \{x_i, y_i\}_{i=1}^{m} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$. In this data set, b (bias) is the value that fixes the offset of the hyper plane ($b \in R$), and w is the normal vector (weight vector) of the hyper plane ($w = (w_1, w_2, \ldots, w_d)^T \in R^d$). Also, m refers to the number of samples in training data, $C$is the regulation coefficient determined by user ($C \in R^+$), x_i is the i$^{\text{th}}$ sample ($x_i = (x_{i1}, x_{i2}, \ldots, x_{id})^T \in R^d$) and y_i is the target value ($y_i \in \{0, 1\} \in R^1$). $\phi(x_i)$ is the function that matches x_i vector to the feature space having a higher size ($\varphi(.)$: $R^n \rightarrow R^m$, $\forall_i \in \{1, 2, \ldots, m\}$). ξ: shows maximum deviation of y_i value from the target value ($\xi_i, \xi_i^* \geq 0$).

SVR function is defined as in (17), which aims to find an optimal hyper plane that tries to separate classes of data points. Optimization problem given in (18-19) is solved according to the equations in (20-21).

$$f(x_i) = w^{\mathrm{T}} \phi(x_i) + b \tag{17}$$

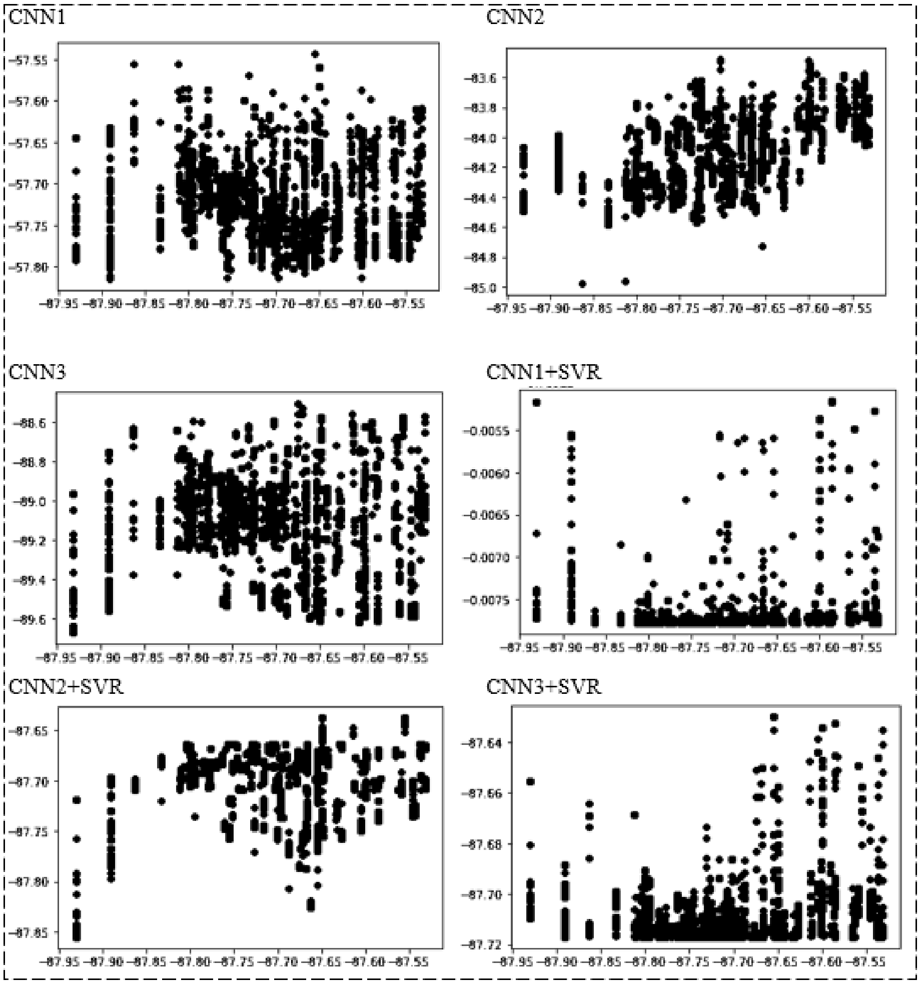

FIGURE 11. Prediction values by CNN models and GPSCNNs for dataset#2.

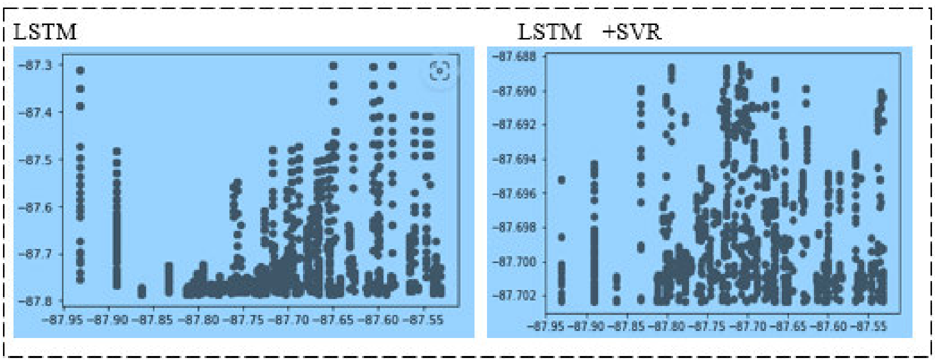

FIGURE 12. Prediction values by GPSLSTM for dataset#2.

$$min_{w\,\epsilon\, R^d,\; b\,\epsilon\, R}\ \left(\frac{1}{2}\ ||w||^2 + C\sum_{i=1}^{m}|\xi|_{\varepsilon}(f(x_i)\text{-}\ y_i)\right) \quad (18)$$

$$\left.\begin{aligned} &min_{w\,\epsilon\, R^d,\; b\,\epsilon\, R}\ \left(\frac{1}{2}\ w^T w + C\ \sum_{i=1}^{m}(\xi_i +\ \xi_i^*)\right) \\ &min_{w\,\epsilon\, R^d,\; b\,\epsilon\, R}\left(\frac{1}{2}\ w^T w +\ C\ \sum_{i=1}^{m}\xi_i +\ C\ \sum_{i=1}^{m}\xi_i^*\right) \end{aligned}\right\} \quad (19)$$

$$w^T\varphi(x_i) + b - y_i \leq \varepsilon + \xi_i \quad (20)$$

$$y_i - (w^T\varphi(x_i) + b) \leq \varepsilon + \xi_i \quad (21)$$

III. EXPERIMENTAL RESULTS

A. IMPLEMENTATION DETAIL

All the experiments are performed on ''python'' platform. ''Keras'' library is used in deep neural network based models and ''sklearn'' library in ML based models. The learning

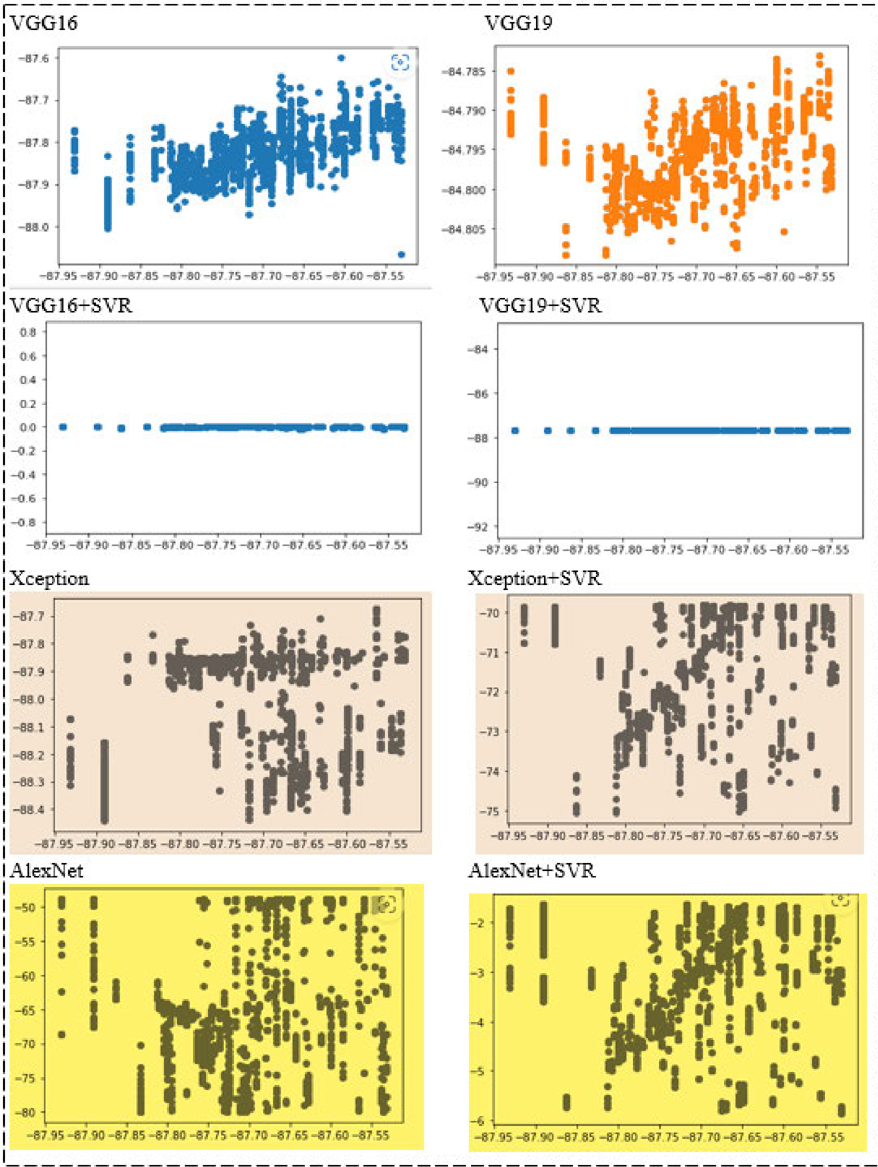

FIGURE 13. Prediction values by GPSCNNs and pre-trained model for dataset#2.

rate is regulated as "0.0001", epoch number as "100", batch size as "150" and optimizer as "Adam". The detailed implementation features of CNN1, CNN2, CNN3 and LSTM deep neural networks are presented in Table 2. Table 3 displays meta-model parameter details in ensemble learning algorithm. The configuration features of the computers used in the study for the models are as follows: Intel(R) Core(TM) i7-7700 processor, ~3.6GHz and 2245676MB, 24GB memory, Intel(r) Hd Graphics 630, NVIDIA GeForce GT 730 graphics card.

B. EVALUATION METRICS

The effectiveness of the proposed models was tested on two different data sets. Training and test data sets were randomly selected as 80% training and 20% test data. The performances of each model were evaluated in terms of mean absolute error (MAE) and mean squared error (MSE) test metrics. Fig. 5 and Table 4 summarize the performance results. The proposed ensemble models provided minimum MAE and MSE values for both data sets, which indicates promising results for the proposed deep hybrid ensemble models. Generally speaking, ensemble (kNN+LR, kNN+DTR, ensemble stacking) and hybrid models CNN1+SVR, CNN2+SVR, CNN3+SVR, VGG16+SVR, VGG19+SVR, Xception+SVR, AlexNet+SVR, LSTM+SVR) were found to be superior to single models (CNN1, CNN2, CNN3, SVR, VGG16, VGG19, Xception, AlexNet, LSTM, kNN, LR, DTR) due to fewer errors. For the first data set, "kNN+DTR" provided a minimum MAE value (0.4622) and "kNN+LR" a minimum MSE value (0.4070). As for the second data set, "ensemble stacking" provided a minimum MAE value (0.0278) and a minimum MSE value (0.003).

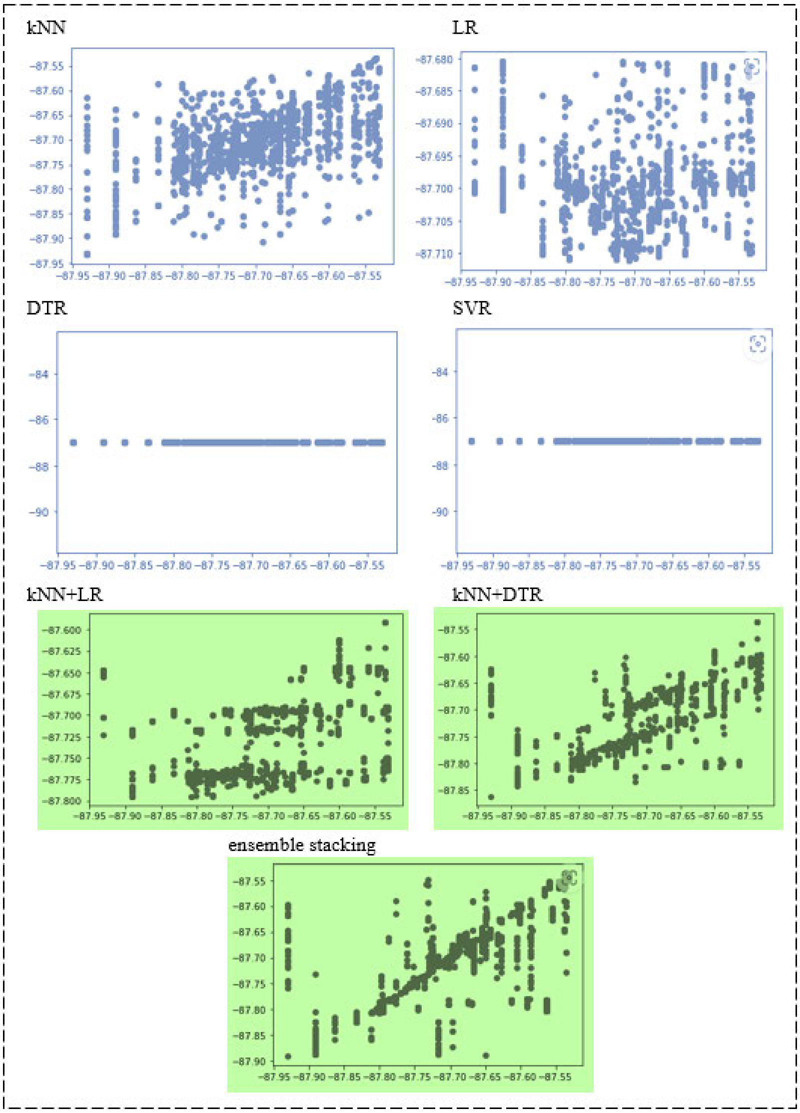

FIGURE 14. **Prediction values by ML and ensemble models for dataset#2.**

The comparison of the longitude values ($\hat{y}$) predicted in Fig. 7-14 to the real longitude values (y) in Fig. 6 shows that hybrid or ensemble models for the first and second data set generally provide more accurate predictions than single models. It was also found that ensemble models provide the most precise positioning.

IV. CONCLUSION

Numerious methods were developed in the past to track people's location and mobility. High accuracy satellite positioning system directly or indirectly lead to many problems in various fields (mobility, balance disorders, deconditioning, team sports, meteorological practices etc.). Highly precise satellite positioning is indispensable in such sensitive circumstances. Real-time improvements in accuracy positioning and navigation technologies can be achieved by analyzing GPS data. In these performance analyzes, sources of error including satellite orbit accuracy, horizontal and vertical accuracy errors, standard deviation andradio delay, and the validity of the data from GPS can be observed.

In addition, meta-analysis methods should be adopted so as to support GPS analysis during land shipping due to the necessity for standardization of fleet speed zones, vagueness of activity descriptors and inequalities in work phase model at micro level. Although positioning is a basic need in many fields, there are still obstacles hindering the advancement of GPS technology. The present study used latitude and longitude points, which are among the built-in functions of GPS systems, to improve positioning accuracy by using GPS data. ML and DL based different models were tested on two different data sets in order to improve positioning performance. These data sets are publicly available online. Coordination conversion was practiced via these models, and effectiveness of coordinate conversion prediction by each model was evaluated. The effect of the train/test split methods was examined and all models were evaluated with test metrics (MAE, MSE). While the proposed ensemble (kNN+LR, kNN+DTR, and ensemble stacking) and hybrid models (GPSCNNs and GPSLSTM) have found to show superior performance; the best ones have proved to be ensemble models. In this context, the experimental results have shown that the proposed ensemble and hybrid models are instrumental in improving the positioning accuracy of navigation systems and are superior to existing positioning-based approaches in the literature.

REFERENCES

[1] A. Noureldin, A. Osman, and N. El-Sheimy, "A neuro-wavelet method for multi-sensor system integration for vehicular navigation," *Meas. Sci. Technol.*, vol. 15, no. 2, pp. 404–412, Feb. 2004.

[2] M. R. Islam and J.-M. Kim, "An effective approach to improving low-cost GPS positioning accuracy in real-time navigation," *Sci. World J.*, vol. 2014, pp. 1–8, Jan. 2014.

[3] K. Krasuski, D. Popielarczyk, A. Ciećko, and J. Ćwiklak, "A new strategy for improving the accuracy of aircraft positioning using DGPS technique in aerial navigation," *Energies*, vol. 14, no. 15, p. 4431, Jul. 2021.

[4] J.-Y. Huang and C.-H. Tsai, "Improve GPS positioning accuracy with context awareness," in *Proc. 1st IEEE Int. Conf. Ubi-Media Comput.*, Jul. 2008, pp. 94–99. [Online]. Available: https://ieeexplore.ieee.org/document/4570872

[5] D. Bhatt, P. Aggarwal, V. Devabhaktuni, and P. Bhattacharya, "A new source difference artificial neural network for enhanced positioning accuracy," *Meas. Sci. Technol.*, vol. 23, no. 10, Oct. 2012, Art. no. 105101.

[6] J. Li, N. Song, G. Yang, M. Li, and Q. Cai, "Improving positioning accuracy of vehicular navigation system during GPS outages utilizing ensemble learning algorithm," *Inf. Fusion*, vol. 35, pp. 1–10, May 2017.

[7] Z. Li, J. Wang, B. Li, J. Gao, and X. Tan, "GPS/INS/odometer integrated system using fuzzy neural network for land vehicle navigation applications," *J. Navigat.*, vol. 67, no. 6, pp. 967–983, Nov. 2014.

[8] X. Chen, C. Shen, W.-B. Zhang, M. Tomizuka, Y. Xu, and K. Chiu, "Novel hybrid of strong tracking Kalman filter and wavelet neural network for GPS/INS during GPS outages," *Measurement*, vol. 46, no. 10, pp. 3847–3854, Dec. 2013.

[9] A. El-Shafie, A. Najah, and O. A. Karim, "Amplified wavelet-ANFIS-based model for GPS/INS integration to enhance vehicular navigation system," *Neural Comput. Appl.*, vol. 24, nos. 7–8, pp. 1905–1916, Jun. 2014.

[10] B. H. Kaygısız, A. M. Erkmen, and İ. Erkmen, "Enhancing positioning accuracy of GPS/INS system during GPS outages utilizing artificial neural network," *Neural Process. Lett.*, vol. 25, no. 3, pp. 171–186, Jun. 2007.

[11] Y. Zhang, C. Shen, J. Tang, and J. Liu, "Hybrid algorithm based on MDF-CKF and RF for GPS/INS system during GPS outages," *IEEE Access*, vol. 6, pp. 35343–35354, 2018.

[12] H. Zhang, T. Li, L. Yin, D. Liu, Y. Zhou, J. Zhang, and F. Pan, "A novel KGP algorithm for improving INS/GPS integrated navigation positioning accuracy," *Sensors*, vol. 19, no. 7, p. 1623, Apr. 2019.

[13] Y. Xiong, Y. Zhang, X. Guo, C. Wang, C. Shen, J. Li, J. Tang, and J. Liu, "Seamless global positioning system/inertial navigation system navigation method based on square-root cubature Kalman filter and random forest regression," *Rev. Sci. Instrum.*, vol. 90, no. 1, Jan. 2019, Art. no. 015101.

[14] W. Wang, "A study of GPS coordinate transformation theory based on the algorithm of K-medoid," *Eng. Surveying Mapping*, vol. 8, pp. 24–28, Jan. 2016.

[15] L. Song and W. Hu, "GPS coordinate transformation based on neural network coordinate difference learning," *Bull. Surveying Mapping*, vol. 1, pp. 29–31, Mar. 2012.

[16] Kaggle. (2018). *California Housing Dataset—ML Crash Course*. Accessed: Jun. 2022. [Online]. Available: https://www.kaggle.com/datasets/vikramtiwari/california-housing-dataset-ml-crash-course?resource=download&select=california_housing_train.csv

[17] GitHub. (2018). *Project-4-GADSI*. Accessed: Jun. 2022. [Online]. Available: https://github.com/rylativity/identifying_wnv_locations-raw_repo-

[18] *Project 4: West Nile Virus Prediction*. Accessed: Jun. 2022. [Online]. Available: https://www.benjamindornel.com/wnv-prediction.html

[19] *Address-Accuracy-Program-Customer-Guide*. Accessed: Jun. 2022. [Online]. Available: https://www.anchorcomputer.com/wp-content/uploads/2016/04/Address-Accuracy-Program-Customer-Guide.pdf

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[22] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 9, 1997, pp. 155–161.

TOLGA AYDIN received the M.Sc. and Ph.D. degrees in computer engineering from Bilkent University, Turkey. He is currently an Assistant Professor of computer engineering with Ataturk University, Turkey. His research interests include algorithms, artificial intelligence, machine learning, and data mining.

EBRU ERDEM received the M.Sc. degree in computer engineering from Ataturk University, Turkey. She is currently a Research Assistant of computer engineering with Ataturk University. Her research interests include computer science, algorithms, distributed systems, artificial intelligence, and engineering and technology.