Shinkuma *et al. EURASIP Journal on Wireless Communications and Networking* (2020) 2020:92
https://doi.org/10.1186/s13638-020-01709-1

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Data assessment and prioritization in mobile networks for real-time prediction of spatial information using machine learning

Ryoichi Shinkuma*, Takayuki Nishio, Yuichi Inagaki and Eiji Oki

*Correspondence:
shinkuma@i.kyoto-u.ac.jp
Graduate School of Informatics, Kyoto University, Yoshida-honmachi, Sakyo-ku, Kyoto, Japan

Abstract

A new framework of data assessment and prioritization for real-time prediction of spatial information is presented. The real-time prediction of spatial information is promising for next-generation mobile networks. Recent developments in machine learning technology have enabled prediction of spatial information, which will be quite useful for smart mobility services including navigation, driving assistance, and self-driving. Other key enablers for forming spatial information are image sensors in mobile devices like smartphones and tablets and in vehicles such as cars and drones and real-time cognitive computing like automatic number/license plate recognition systems and object recognition systems. However, since image data collected by mobile devices and vehicles need to be delivered to the server in real time to extract input data for real-time prediction, the uplink transmission speed of mobile networks is a major impediment. This paper proposes a framework of data assessment and prioritization that reduces the uplink traffic volume while maintaining the prediction accuracy of spatial information. In our framework, machine learning is used to estimate the importance of each data element and to predict spatial information under the limitation of available data. A numerical evaluation using an actual vehicle mobility dataset demonstrated the validity of the proposed framework. Two extension schemes in our framework, which use the ensemble of importance scores obtained from multiple feature selection methods, are also presented to improve its robustness against various machine learning and feature selection methods. We discuss the performance of those schemes through numerical evaluation.

Keywords: Spatial information, Real-time prediction, Mobile crowdsensing, Data assessment, Machine learning, Feature selection

1 Introduction

The demand for real-time prediction of spatial information is steadily growing [1]. Spatial information includes traffic flows (cars, pedestrians, etc.), road surface conditions, construction activities, noise levels, air quality, traffic-related accidents, and crimes. The latest Geospatial Industry Outlook and Readiness Index report (GeoBuiz-18) estimates that the geographic information system and spatial analytics market will grow from 66.2 billion U.S. dollars in 2017 to 88.3 billion in 2020 [2].

© The Author(s). 2020 **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

The development of real-time cognitive computing is essential to providing such real-time prediction of spatial information; most of the information mentioned above can be extracted from still or moving images acquired by cameras. For instance, automatic number-plate/license plate recognition enables the locations of individual vehicles to be determined [3], from which information on vehicle traffic flows can be obtained. Recent advances in object recognition technology have enabled data to be collected for specific target objects, including people (either walking or standing), damaged road surfaces, and car accidents. This provides a powerful way of obtaining input data for predicting spatial information.

Other key-enabling technologies are those found in small, lightweight, energy-saving image sensors that can produce high-quality fine-grained images. The reason "high quality" is important is that the performance of the recognition technologies mentioned above depends on the quality of the input images. The reason image sensors must be "small, lightweight, and energy saving" is that they have to be implemented in small devices like smartphones, drones, and wearable devices. According to MarketsandMarkets, the image sensor market is expected to grow from 12.8 billion U.S. dollars in 2016 to 24.8 billion by 2023 [4].

A question we thus need to address is, "How can we collect input data for predicting spatial information?" If we relied only on fixed image sensors, we would need a number of dense grids of fixed image sensors to cover a wide geographical area like New York City or Tokyo, which would be inefficient in terms of deployment and maintenance costs. Mobile crowdsensing (MCS) has been proposed to effectively collect sensor data [5]. In MCS, conventional mobile devices equipped with sensors such as smartphones as well as platforms that are becoming mobile devices, such as vehicles equipped with cameras and sensors, work as distributed mobile sensors to acquire sensor data associated with their respective locations in a timely manner. Thanks to the inherent nomadic characteristic of mobile device users, this approach should provide spatially and temporally complete coverage of wide geographical areas. Thus, the integration of fixed sensors and MCS is also a key for collecting image sensor data.

However, since image data collected by mobile devices and vehicles need to be delivered to the server in real time to extract input data for real-time prediction, there is an obstacle that must be overcome: the relatively slow uplink transmission in mobile networks. According to the officially announced specifications of Long-Term Evolution-Advanced, the highest speed (peak rate) is 1.5 Gbps. The 5th generation network being developed will have a peak rate of up to 10 Gbps. However, actual throughput is generally much lower than the peak rate (possibly as low as a few of a percent of the peak rate) because the peak rate is the rate measured under ideal conditions. Under real-world conditions, particularly in the uplink, transmission speed is limited due to multiple deterioration factors like the poor transmission capability of mobile devices and the contention-based access of wireless devices. In contrast, the data rates of image sensors can be quite high, potentially of the order of billions of bits per second [6]. Obviously, if many mobile and fixed sensors connected to a base station transmit image sensor data through the uplink at the same time, the total amount of transmitted data could easily exceed the capacity of the uplink.

This paper proposes a new framework to overcome this uplink bottleneck problem:

the use of assessment and prioritization of image data collected by mobile devices from which input data for real-time prediction are extracted. Assessment means assessing the importance of image data collected by each mobile device. Prioritization means assigning a higher priority to the more important image data in uplink transmission. Prediction has to be performed only using the available data because some data will be missing because of the limited capacity of uplink transmission. We evaluated the effectiveness of this approach by using an actual vehicle mobility dataset. The use of machine learning for data assessment and prioritization drastically reduced the uplink traffic volume while maintaining prediction accuracy.

Data in the proposed framework are assessed by using feature selection of machine learning [7, 8]. Feature selection enables the importance score of each data element to be extracted from the prediction model of machine learning. However, the score is not robust against various machine learning and feature selection methods: different feature selection methods may produce different importance scores for the same data element. To solve this problem, this paper presents two extension schemes in the proposed framework, which use the ensemble of importance scores obtained from multiple feature selection methods. Those schemes are validated through numerical evaluation. Note that this paper is the extended version of our previous paper [9].

The rest of this paper is organized as follows. Section 2 introduces the related work, which includes an overview of the prior works on feature selection. Section 3 presents the system model, the basic idea, and the problem formulation of our data assessment and prioritization framework. We then present and discuss the results of our numerical evaluation in Section 4. Then, the extension schemes with numerical results are presented in Sections 5 and 6. Finally, Section 7 concludes with a summary of the key points and a mention of future work.

2 Related work

2.1 Prioritization methods

In traditional mobile networks, delay-sensitive data for interactive applications like teleconferences or online games have been widely given higher priority to improve user experience quality [10]. However, in next-generation mobile networks, communication quality for machines will become more important than that for people [11]. The real-time spatial information considered here could be a typical application for machines; it could be used by smart mobility services including navigation, driving assistance, and self-driving. The data assessment and prioritization framework proposed in this paper is well suited for such machine centric applications in next-generation mobile networks. Similar works have been done by other researchers including some of the authors of this paper [12, 13]. Yamada et al. presented software-defined network (SDN)-based control for mobile traffic prediction using past traffic logs collected at base stations. Inagaki et al. presented Internet-of-Things (IoT) device control to predict spatial information in real-time. The main differences between this work and those works are as follows: (1) this work considers and compares multiple kinds of machine learning methods, whereas the other works considered only random forest (RF) as a machine learning method, and (2) this work discusses the ensemble of importance scores obtained by multiple feature selections, whereas the other works did not.

2.2 Feature selection methods

2.2.1 Overview

Gevrey et al. systematically discussed methods for evaluating the contributions of data to prediction [7]. Methods for evaluating the contributions of data are commonly referred to as "feature selection" methods. According to Chandrashekar and Sahin [8], feature selection methods can be categorized as filter selection, wrapper selection, and embedded selection. Filter selection is done during pre-processing: the correlation or similarity of data is analyzed in advance before machine learning. Wrapper selection is more like an optimization approach: it attempts to find the combination of elements that maximizes prediction accuracy. Embedded selection is the most practical and convenient: it enables the importance of each element to be determined through a machine learning training process.

2.2.2 Commonly used methods

The perturb method aims to assess the effect of small changes in each input on the output in machine learning [7]. The algorithm adjusts the input values of one variable while leaving the others untouched. The change in the output result for each change in the input elements is recorded. The mean squared error (MSE) of the output should increase as a larger amount of noise is added to the important input variable. This method is straightforward and applicable to a wide variety of machine learning methods including multilayer perceptron (MLP) and RF though it does not belong to any of the three categories suggested by Chandrashekar and Sahin.

The weights method uses an embedding approach. It essentially involves partitioning the hidden-output connection weights of each hidden neuron into components associated with each input element [7]. It can be applied to neural-network (NN)-based methods including a long short-term memory (LSTM) network, which will be explained later in Section 4.

The impurity method also uses an embedding approach. The RF machine learning algorithm is highly accurate and far more robust than decision trees. It can model a huge feature space [14]. In RF, at each node of the decision tree, m elements are randomly selected out of the total number of features, and the best split is selected out of these m elements. At each node t within the binary trees T of RF, the optimal split is sought by using the impurity $i(t)$ [15], which is a computationally efficient approximation of the entropy—measuring how well a potential split separates the samples of the two groups in this particular node. This means that the impurity reflects the importance of the element used for splitting.

2.2.3 Ensemble methods

According to Shen et al. [16], feature selection ensemble is an ensemble-based method that aims to construct a group of feature subsets and then produce an aggregated result out of the group. In so doing, the performance variance of obtaining a single result from a single approach can be reduced. It is also intuitively appealing that the combination of multiple subsets may remove fewer important features, resulting in a compact, robust, and efficient solution.

Saeys et al. introduced the use of ensemble methods for feature selection [17]. They showed that by constructing ensemble feature selection techniques, robustness of feature

ranking and feature subset selection could be improved by using similar techniques as in ensemble methods for supervised learning. When analyzing robustness versus classification performance, ensemble methods show great promise for large-feature/small-sample-size domains. It turns out that the best trade-off between robustness and classification performance depends on the dataset at hand, giving rise to a new model selection strategy, incorporating both classification performance as well as robustness in the evaluation strategy.

3 Proposed framework

3.1 System model

Figure 1 illustrates a system for predicting spatial information in real time. Mobile devices equipped with a sensor acquire sensor data and transmit them to a base station through an uplink channel. Here, we assume that vehicles equipped with image sensors (cameras) are the mobile devices. The base stations forward the received data to a server via a relay network. The server extracts the elements of spatial information from the collected sensor data and produces spatial information.

Figure 2 shows a block diagram of the system. Each mobile device consists of a sensor, pre-processor, transmitter, controller, and data storage, while the server consists of a receiver, converter, learner, predictor, and evaluator. In this figure, the solid lines indicate data flows, while the broken lines indicate control messages.

Image data collected by mobile devices need to be delivered to the server in real time to extract input data for real-time prediction even when the transmission speed is low due to bandwidth limitation, uplink traffic congestion, or poor signal quality. Therefore, under such bandwidth limitations, only mobile devices with higher priority data are allowed to

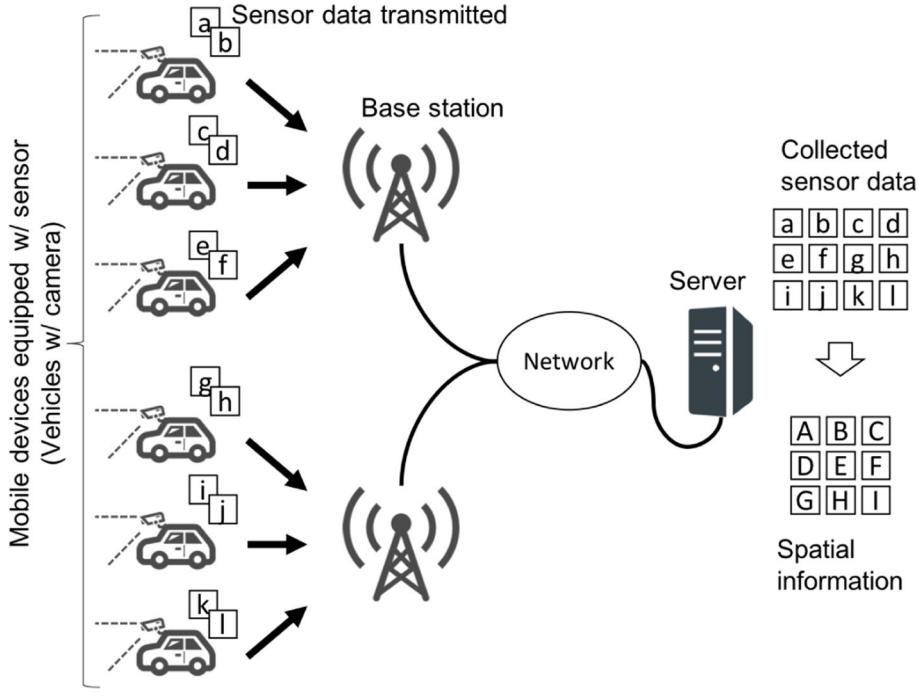

Fig. 1 System for predicting spatial information in real time

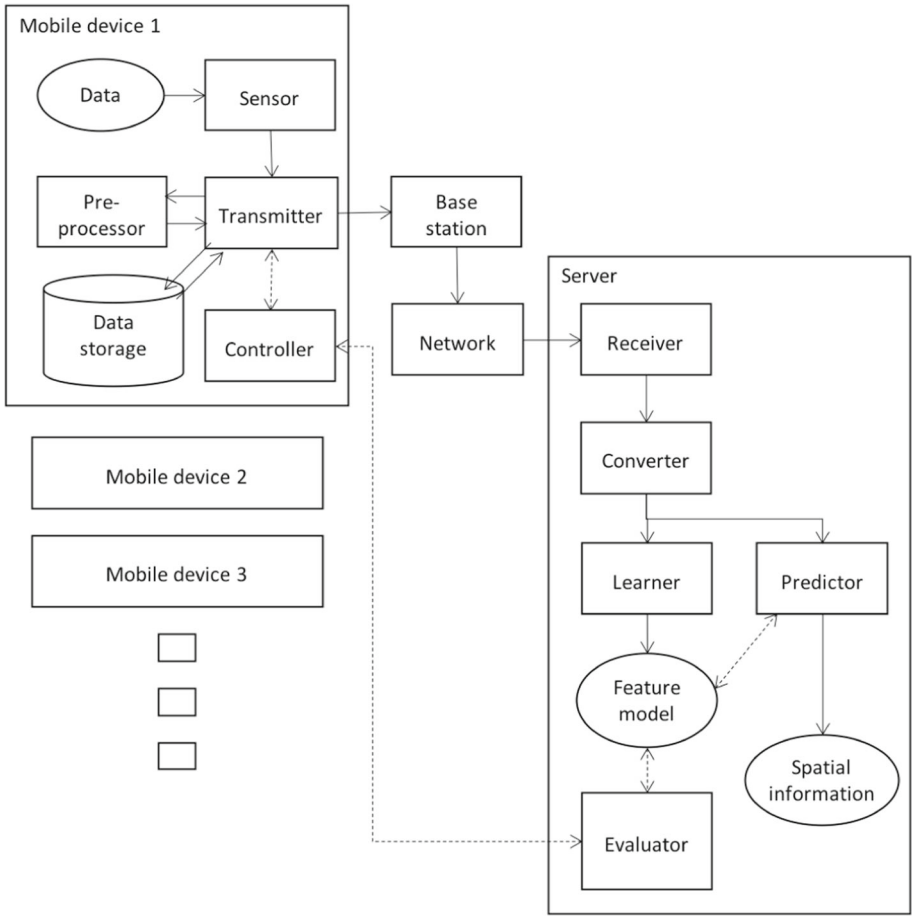

Fig. 2 Block diagram of proposed framework. Solid and broken lines indicate data flows and control messages, respectively

transmit their image data. The controller prioritizes the sensor data in accordance with the importance of each element of spatial information as determined by the evaluator. The server receives the sensor data and converts them into a form that can be used by the learner as input data for prediction. The predictor in the server predicts and produces spatial information by using the feature model, which will be discussed later in Section 3.2.

In contrast, training data for prediction can be collected from mobile devices as a background process. That is, the way to forward them to the server is out of the scope of our framework because they can be forwarded through mobile networks at off-peak traffic time or a rich bandwidth provided by WiFi or millimeter-wave transmission. The learner in the server produces a feature model by using the training data received through the machine-learning training process discussed in Section 3.2.

Figure 3 illustrates the process flow of the proposed framework, which is split into the rea time and background flows. In this figure, the solid lines indicate the main flow of the process, while the broken lines indicate the information update. In the real-time flow, first, sensor data are acquired at each mobile device. Then, the data are prioritized at mobile devices on the basis of the importance given from the background flow. Data are transmitted by mobile devices in accordance with the priority. Here, we can consider two

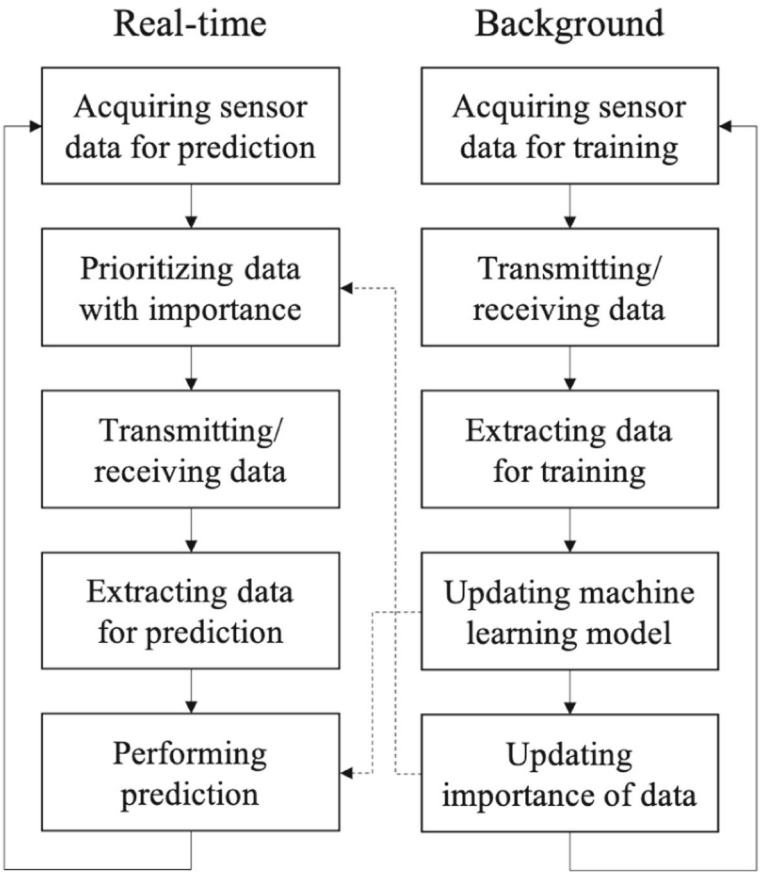

Fig. 3 Process flow in proposed framework. Solid and broken lines indicate main process flow and information update, respectively

cases: (1) communication capacity can be estimated and (2) communication capacity is not given. In the former case, an existing method for capacity estimation in communication networks would be used. A simple and classical approach for this is measuring roundtrip time, as suggested in prior works [18–20]. If communication capacity can be successfully estimated in advance, transmitted data are limited before actually being transmitted, so the total volume of the transmitted data does not exceed the communication capacity. In the latter (no given communication capacity) case, transmitted data are dropped by the channel access control protocol, so the total volume of the transmitted data does not exceed the communication capacity as actually occurs in real systems such as wireless local area networks and cellular networks. In both cases, in our framework, data with high importance are transmitted with high priority. At the server, data are extracted and are used as input for prediction. In the background flow, sensor data for training are acquired at each mobile device. As we mentioned above, the way to prioritize and transmit data for training is out of the scope of this paper. Data used as input and output for training are extracted at the server. The ML model is updated using those data, and the updated model is used for performing prediction in the real-time flow. The importance of data is also updated, and it is used when data are prioritized for transmission in the real-time flow.

3.2 Prediction and data assessment using machine learning

Figure 4 shows three example patterns of spatial information recorded on different days: d_1, d_2, and d_3. They show the values, which could indicate the volume of any spatial information extracted from image sensor data like the volume of road traffic (the numbers of vehicles or pedestrians), for five geographical sections (sections A to E) for each time slot (1:01 to 1:05 pm). Each pattern consists of a set of input data and the expected result. The learner accumulates the recorded patterns. The predictor predicts the future results, which will be actually obtained at 1:00pm, from the currently obtained input data if it finds the corresponding input data in the recorded patterns.

In the figure, the black elements are common to the three patterns, while white elements vary among the patterns. This means that the white elements are meaningful for prediction. In other words, the white elements are more important for prediction than the black ones.

However, there are two problems. The exact same input is rarely found in the recorded patterns, so the prediction needs to be done using similar previously recorded inputs. Elements in different records that are exactly the same are also rarely found, so the importance of elements needs to be evaluated over different records even though they are not exactly the same. These two problems are overcome by machine learning of data features.

By using a sufficient number of recorded patterns as training data, supervised learning using an NN or RF method produces generalized feature models that enable the system to perform prediction from the immediately acquired input even if the exact same input is not found in the recorded patterns. Moreover, machine learning enables the system to evaluate which elements are important for prediction. As we explained in Section 2.2, this

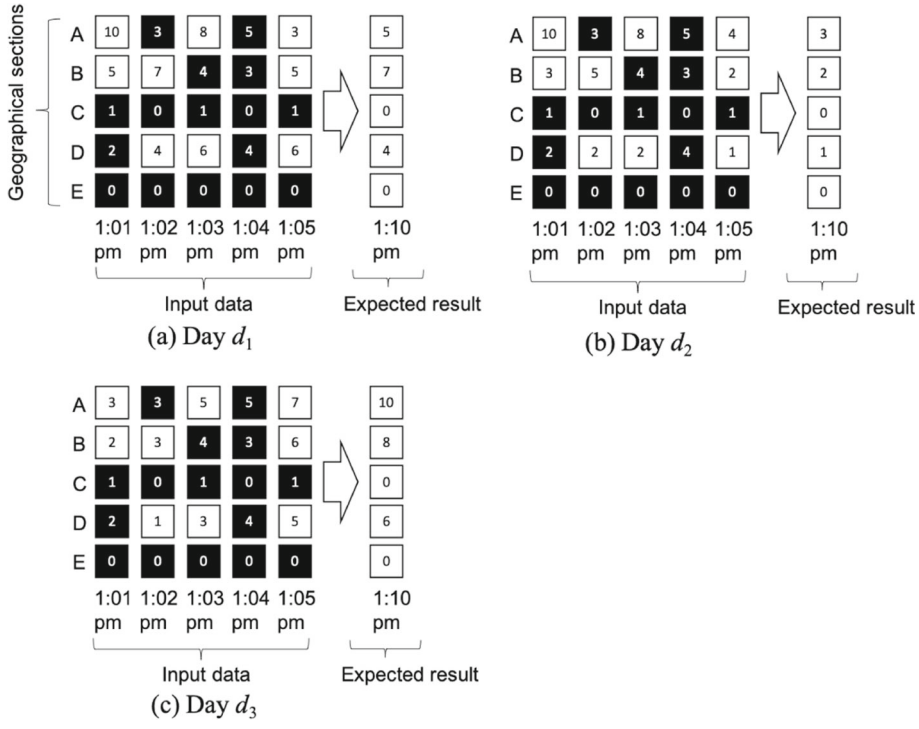

Fig. 4 Example patterns of spatial information recorded on different days

capability is called feature selection and enables us to obtain the importance score of each data element.

3.3 Formulation of proposed framework

This section presents the key idea of our framework. The objective function and traffic-volume constraint of our framework can be formulated as

$$\max_{X(t)} A(X(t)), \tag{1}$$

$$\sum_{x \in X(t)} d_x \leq C(t), \tag{2}$$

where $X(t)$ and $A(X(t))$ mean the set of input data received from mobile devices for prediction at time t and the accuracy of the prediction at time t achieved using $X(t)$, respectively. In Eq. (2), d_x and $C(t)$ mean the data volume of an input data element x and the capacity of the network at time t, respectively. Equation (2) is the constraint meaning that the total volume of data transmitted by mobile devices must be smaller than or equal to the capacity of the network. However, since the prediction system is operated on a real-time basis, it is impossible to search for and find the optimal $X(t)$ among all possible sets of $X(t)$. Therefore, in our framework, we convert Eq. (1) into

$$\max_{X(t)} \sum_{x \in X(t)} s_x, \tag{3}$$

where s_x means the importance score of input data element x obtained by using a feature selection method. The constraint in Eq. (2) still works for Eq. (3). Equation (3) means that we need to maximize the total score of input data for prediction. Converting Eq. (1) into Eq. (3) is reasonable because, as we mentioned above, feature selection methods give higher scores to input data that make larger contributions to prediction accuracy.

Finally, we mention how to solve Eq. (3). This problem is considered as a classical 0-1 Knapsack problem [21]. A simple approach for this is just to sort x in the ascending order of s_x, and then "greedily" pick as many x as possible from the top under the constraint of (2).

4 Numerical evaluation

4.1 Settings

We numerically evaluated the validity of our framework. We considered prediction of car traffic volume in a specific geographical area as our evaluation scenario.

The same as Wang et al. [5], we used actual data from the CRAWDAD dataset. We used the global positioning system (GPS) coordinates collected from 536 taxi cabs in San Francisco over 25 days. The unit time of the dataset was 1 min. The dataset was split into 20 days (28,800 min) and 5 days (7200 min) for training and testing, respectively. Twenty percent of the training dataset (5769 min) was used for validation. We defined 144 geographical sections, each 1 km by 1 km, in the San Francisco area. We assumed the presence in each geographical section of at least one device that aggregates raw image sensor data acquired within the geographical section. Therefore, 144 aggregated images were collected over the whole geographical area. The capacity of the uplink transmission was given by the number of aggregated images the server can receive normalized by 144.

We assumed that the number of taxies in each section corresponds to the car traffic volume there and was ideally extracted from the image uploaded from the section. The number of taxies in each section in a sequence of timeslots $t-W, t-W+1, \ldots, t-1$ was used to predict the number of taxies in each section at a time $t+T$. Specifically, results are shown for the case $W = 10$ and $T = 10$. We assumed that training had been done using all data for that in advance before real-time prediction. On the other hand, real-time prediction was performed using available data limited by the capacity of the uplink transmission. When the input data for a section is missing, 0 was used instead for that section.

For machine learning methods for prediction, we used three NN-based methods: MLP, 3D-convolutional NN (3D-CNN) [22], and LSTM from the Keras with Tensorflow. 3D-CNN is an extension of a CNN and well suited for predicting spatiotemporal information. LSTM is a recurrent NN (RNN) algorithm that is well suited for prediction from time series data. The objective of LSTM networks is to model long-term dependencies and determine the optimal time lags for time series problems [23]. These characteristics are especially desirable for short-term prediction due to the lack of prior knowledge of the relationship between prediction results and the length of input historical data. A typical LSTM network has one input layer, and one recurrent hidden layer for which the basic unit is a memory block instead of a traditional neuron node, and one output layer. The memory blocks are a set of recurrently connected subnets. Each block contains one or more self-connected memory cells and multiplicative units: the input, output, and forget gates, which provide continuous analogs of write, read, and reset operations on the cells. We also used RF from the Scikit-learn. Keras and Scikit-learn are a framework and a toolset for machine learning, respectively. Both are well known and widely used. The parameter settings for the three NN-based methods are listed in Table 1. On the other hand, for RF, the number of estimators and the maximum depth were 20 and 20. The MSE was used as the criterion. The other parameters were set in accordance with the default settings of Scikit-learn. For feature selection methods, we used the perturb method for all four machine learning methods. The weights and impurity methods are used only for LSTM and RF, respectively. Data importance obtained by using these feature selection methods was used for data prioritization; for instance, when the capacity of the uplink transmission is 0.5, only the most important 72 images out of 144 can be received by the server.

The NN-based methods and RF are suitable because, in the system model explained in Section 3.1, prediction is performed at the server. If we consider another system model in which prediction is performed at mobile devices, these methods might be too computationally heavy to be deployed because mobile devices are resource-limited. To bridge this gap, lightweight machine learning tailored for edge devices or edge artificial-intelligence methods should be considered for such a system model [25].

4.2 Results

Figures 5 and 6 show the prediction errors as a function of the uplink capacity. The root mean squared logarithmic error (RMSLE) was used as the metric for prediction accuracy [26]. The uplink capacity (shown on the horizontal axes) was defined as the ratio of the number of images received by the server through the uplink to the total number of images generated by the image sensors. Basically, as the uplink capacity decreases, the RMSLE

Table 1 Parameter settings for neural-network-based methods

Batch size	50
Epochs	30
Optimizer	Adam [24]
Learning rate	0.001 [24]
Loss function	Mean absolute error
MLP	Input (no. of units=1440)
	Batch normalization
	Dense (no. of units=512)
	Batch normalization
	Activation (ReLU)
	Dense (no. of units=256)
	Batch normalization
	Activation (ReLU)
	Dense (no. of units=144)
3D-CNN	Input (shape=(10,12,12))
	Conv3D (filters=32, kernel_size=(3,3,3), padding='same')
	Batch normalization
	Activation (ReLU)
	AveragePooling3D (pool_size=(2,2,2), padding='same')
	Conv3D (filters=32, kernel_size=(3,3,3), padding='same')
	Batch normalization
	Activation (ReLU)
	Dense (144)
LSTM	Input (shape=(10,144))
	LSTM (no. of units=128)
	LSTM (no. of units=64)
	Dense (no. of units=144)
Other parameters	Keras default settings

increases. The results for the NN-based methods (MLP, 3D-CNN, LSTM with the perturb method, and LSTM with the weights method) are plotted in Fig. 4, while the results for RF with the perturb method and the impurity method are plotted in Fig. 5. Also plotted are the results for each prediction method with random dropping, which is reasonable as a benchmark because, in data transmission systems without any prioritization, data are randomly dropped when the data rate exceeds the capacity.

As shown in Fig. 5, for MLP, 3D-CNN, and LSTM, data prioritization using the perturb method reduced the uplink capacity to approximately 0.3 while ensuring the best prediction accuracy in the random case. The random case achieved the best prediction accuracy only when all the data were available (uplink capacity = 1.0). LSTM with the weights method also worked well, as did the perturb method. As shown in Fig. 6, for RF, the perturb and impurity methods similarly worked well; both reduced the uplink capacity from approximately 0.1 while ensuring the best prediction accuracy in the random case. These results demonstrate the validity of our framework: data prioritization using feature selection can reduce traffic volume in the uplink while maintaining the quality of spatial information (prediction accuracy in our numerical evaluation).

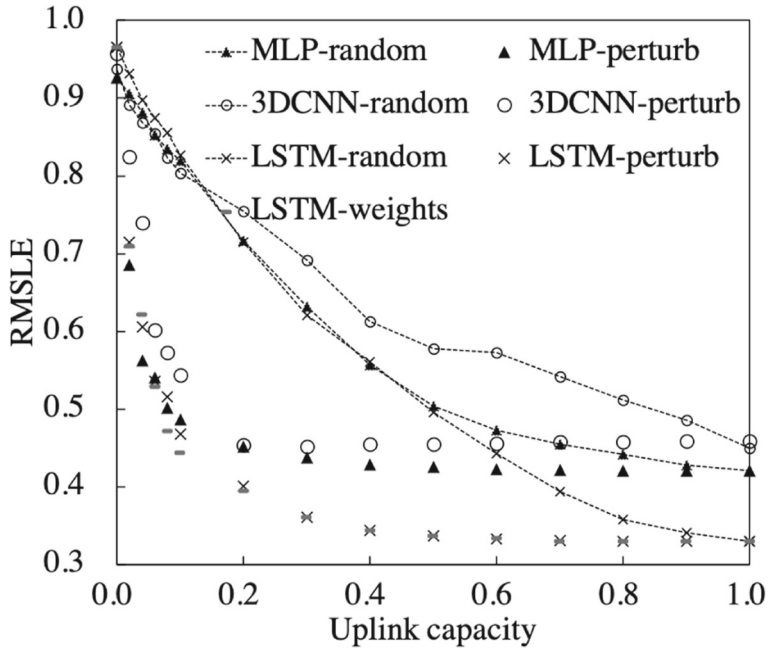

Fig. 5 Prediction accuracy (RMSLE) vs. uplink capacity obtained using neural network-based methods (MLP, 3D-CNN, and LSTM) for prediction. "Perturb" or "weights" method was used for feature selection

5 Extension scheme 1: importance score ensemble

We have validated that data prioritization using feature selection can reduce traffic volume while maintaining the prediction accuracy of spatial information. However, the importance score of data was not robust against various machine learning and feature selection methods: different feature selection methods produced different importance

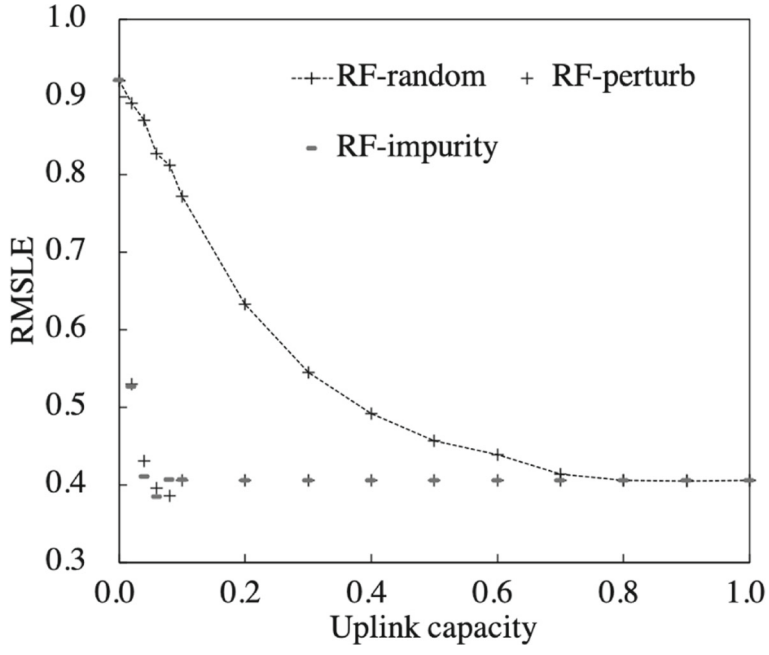

Fig. 6 Prediction accuracy (RMSLE) vs. uplink capacity obtained using RF machine learning method for prediction. Perturb or impurity method was used for feature selection

scores for the same data element. To solve this problem, Sections 5 and 6 present two extension schemes in the proposed framework, which use the ensemble of importance scores obtained from multiple feature selection methods.

5.1 Scheme

As we presented in Eq. (3), s_x represents the importance score of input data element x obtained using a feature selection method in Section 3. Here, by introducing the idea of the feature selection ensemble, which was mentioned in Section 2, we extend s_x as below:

$$s_x = \sum_{f \in \mathrm{F}} g^f \left(s_x^f \right), \tag{4}$$

where f and F mean a feature selection method and a set of feature selection methods used in the proposed framework, respectively.

s_x^f means the importance score of input data element x obtained using feature selection method f.

$g^f()$ is a function that adjusts the scale of importance scores obtained by feature selection method f so that importance scores obtained by different feature selection methods become comparable. This paper considers two options for $g^f()$: ranking and quantizing. The former sorts the importance scores obtained using a feature selection method in ascending order and converts the rank of each importance score into the score. This is inspired by the methods developed by Olsson et al. and Wang et al. [27, 28]. The latter first picks the maximum and minimum of importance scores and assigns them to the top and bottom steps. It then allocates the other scores to the corresponding step between the top and bottom on the basis of the predetermined step size of quantization. This approach is essentially equivalent to the min-max normalization, but the scores are quantized to integers. Note that in ranking, when two importance scores are identical or very similar, they are converted into different ranks, while in quantizing, they are allocated to the same step.

5.2 Numerical evaluation

This section presents the numerical evaluation of extension scheme 1. The setup of the numerical evaluation is basically the same as in Section 4. Figures 7, 8, and 9 plot the results obtained using LSTM, MLP, and RF, respectively. As feature selection methods, the perturb and weights methods were used for LSTM and MLP, while the perturb and impurity methods were used for RF. In the figures, Ext. 1 (rank) and (quant) mean extension scheme 1 with ranking and quantizing, respectively. The number of steps in the quantization was set to 30.

First, Fig. 7 suggests that there were only trivial differences among the methods when we used LSTM. However, in Fig. 8, the perturb method has the best prediction accuracy, while the weights method has the worst. In Fig. 7, the perturb method has the best prediction accuracy again, while the impurity method has the worst. In both figures, extension scheme 1 with ranking and quantizing followed the perturb method and performed much better than the worst methods. This suggests that extension scheme 1, which uses the ensemble of importance scores obtained from multiple feature selection methods, does not always perform best but works robustly as we expected.

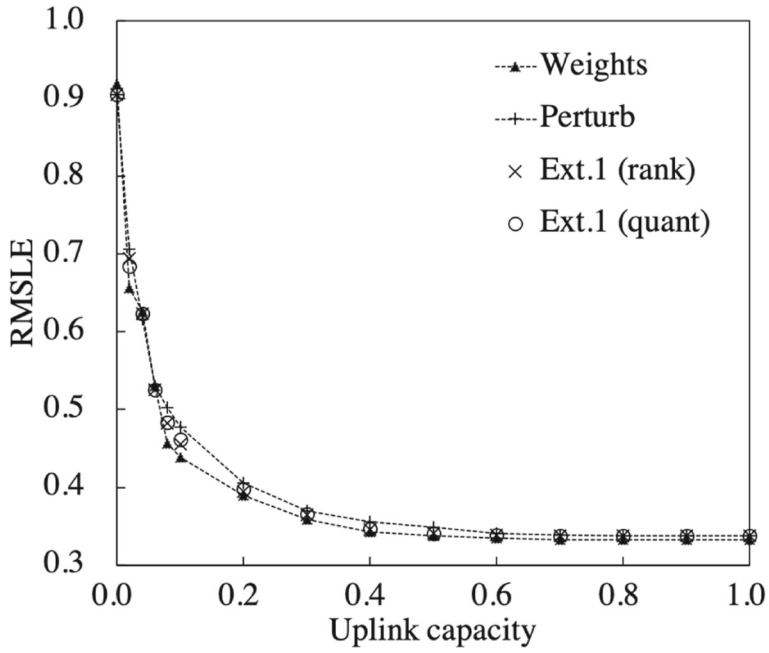

Fig. 7 Prediction accuracy (RMSLE) vs. uplink capacity obtained using LSTM for prediction. Perturb, weights, or extension scheme 1 with ranking and quantizing was used for feature selection

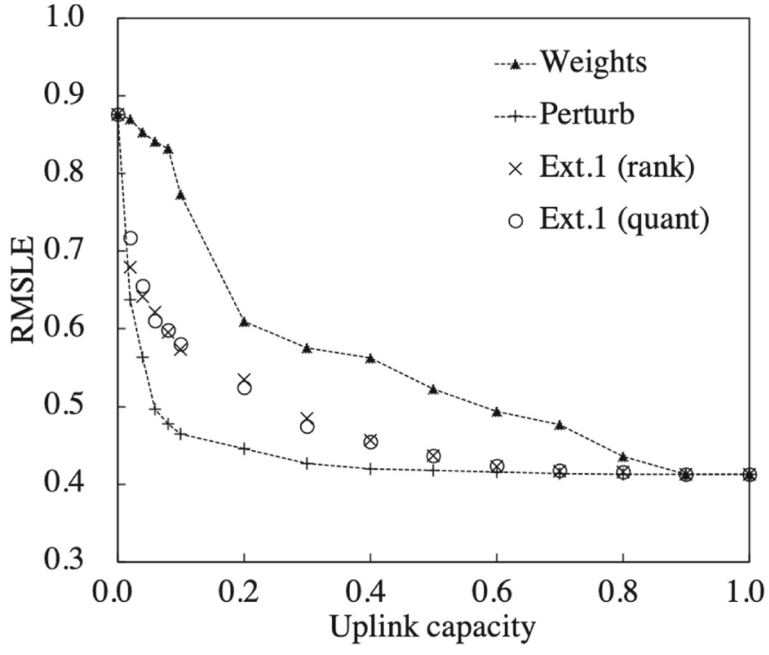

Fig. 8 Prediction accuracy (RMSLE) vs. uplink capacity obtained using MLP for prediction. Perturb, weights, or extension scheme 1 with ranking and quantizing was used for feature selection

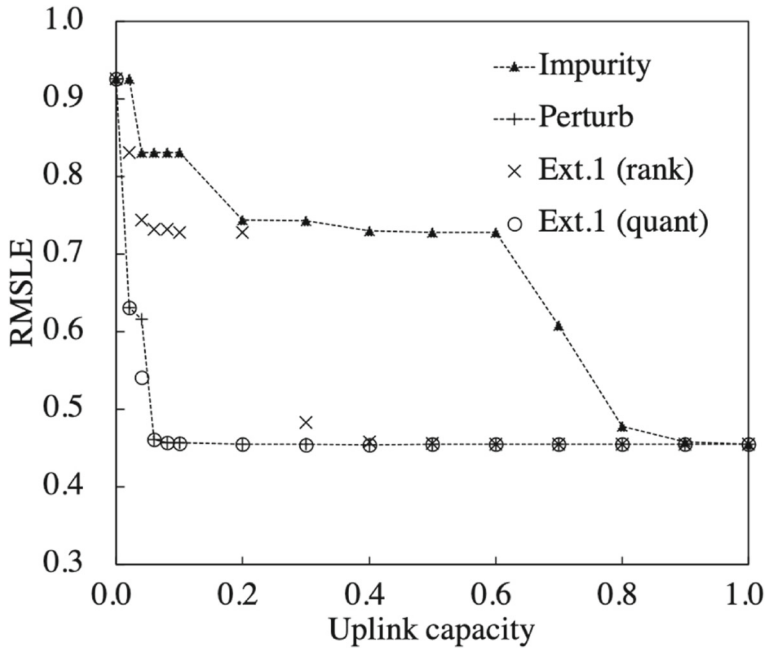

Fig. 9 Prediction accuracy (RMSLE) vs. uplink capacity obtained using RF for prediction. Perturb, impurity, or extension scheme 1 with ranking and quantizing was used for feature selection

6 Extension scheme 2: weighted importance score ensemble

6.1 Scheme

As presented in Section 5, extension scheme 1 sums up importance scores obtained by multiple feature selection methods equally as defined in Eq. (4). Different from this scheme, extension scheme 2 considers the effect of the importance scores obtained by multiple feature selection methods, which is defined as:

$$s_x = \sum_{f \in \mathbb{F}} w^f g^f(s_x^f), \tag{5}$$

where w^f is the weight for the importance score obtained by feature selection method f. This paper considers the prediction accuracy using each feature selection as w^f. More concretely, we measure root mean squared error (RMSE) of prediction for each feature selection method for possible uplink capacities by using the training dataset obtained beforehand and use the inverse of RMSE as the weights in Eq. (5).

6.2 Numerical evaluation

This section demonstrates the numerical evaluation of extension scheme 2. The setup of the numerical evaluation was the same as in Section 4. We particularly want to observe the improvement by extension scheme 2 against extension scheme 1. Note we used ranking for extension scheme 1.

Figures 10, 11, and 12 show the results where LSTM, MLP, and RF were used, respectively. Ext. 1 and Ext. 2 mean extension schemes 1 and 2, respectively. In Fig. 10, we see the same observation as in Fig. 7; the methods differ only slightly. In Figs. 11 and 12, extension scheme 2 outperforms extension scheme 1, which performs similarly to the perturb

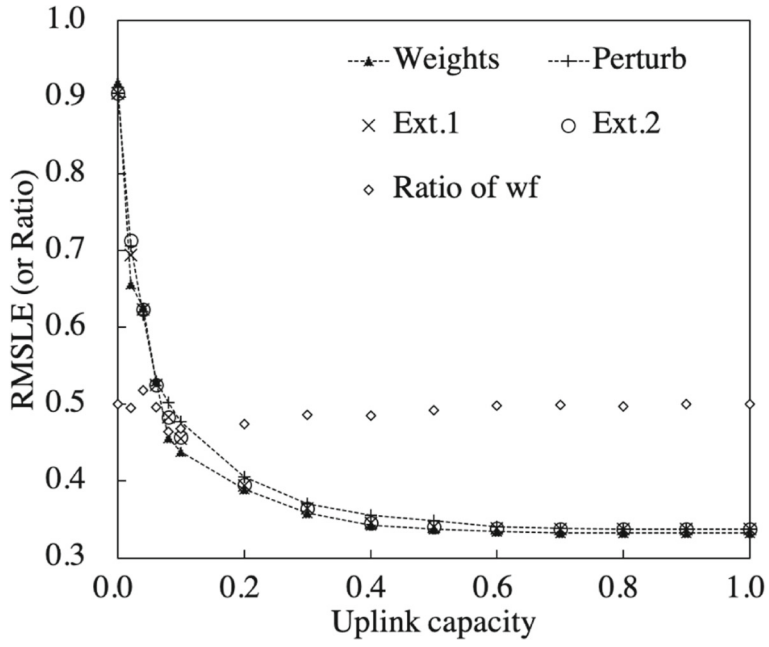

Fig. 10 Prediction accuracy (RMSLE) vs. uplink capacity obtained using LSTM for prediction. Perturb, weights, extension scheme 1 with ranking, or extension scheme 2 were used for feature selection. Ratio of w^f for perturb to other in extension scheme 2 is also plotted

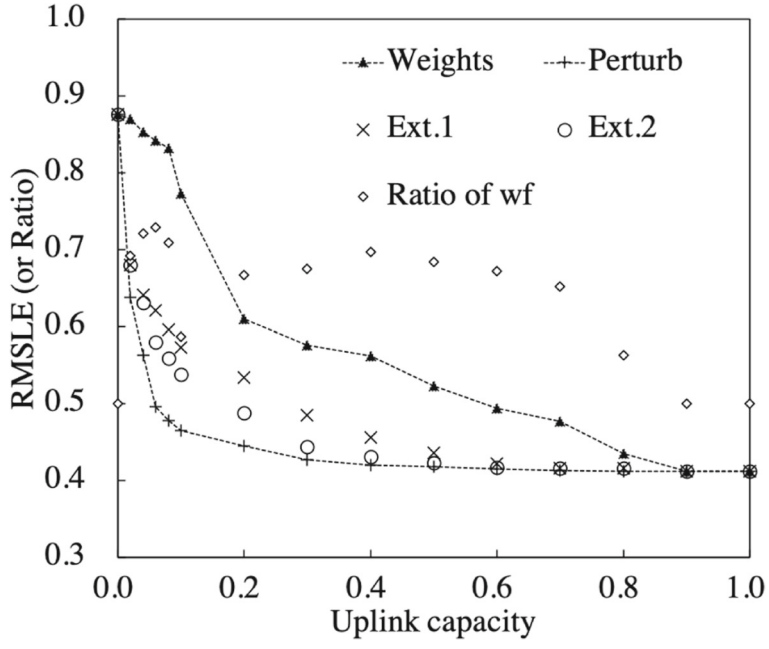

Fig. 11 Prediction accuracy (RMSLE) vs. uplink capacity obtained using MLP for prediction. Perturb, weights, extension scheme 1 with ranking, or extension scheme 2 were used for feature selection. Ratio of w^f for perturb to other in extension scheme 2 is also plotted

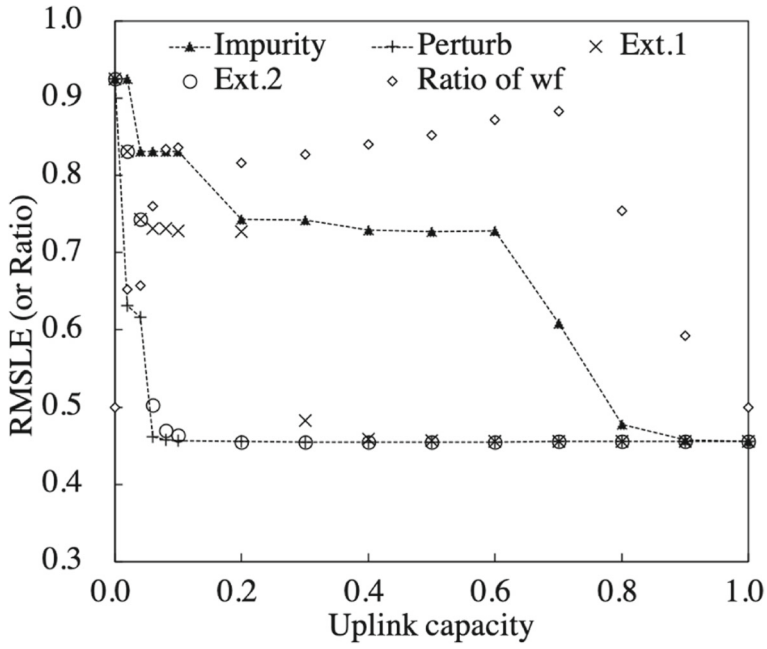

Fig. 12 Prediction accuracy (RMSLE) vs. uplink capacity obtained using RF for prediction. Perturb, impurity, extension scheme 1 with ranking, or extension scheme 2 were used for feature selection. Ratio of w^f for perturb to other in extension scheme 2 is also plotted

method. This result proves that introducing the weight as defined in Eq. (5) effectively improves the prediction accuracy against extension scheme 1.

We also plotted the ratio of w^f in Eq. (5) for the perturb method to the one for the other method in Figs. 10 to 12. In Fig. 10, the ratio of w^f for the perturb method is around 0.5 because the perturb method and the other method (weights) performed similarly. However, in Figs. 11 and 12, the ratio of w^f for the perturb method changes in accordance with how much the perturb method contributed to the improvement of prediction accuracy compared with the other method (weights or impurity). Through the above observation, we have confirmed that extension scheme 2 works as we expected.

7 Conclusion

This paper addressed the problem of reducing the uplink traffic volume in mobile networks while maintaining the accuracy of the spatial information prediction under the limitation of available data. Our machine-learning-based approach is based on evaluating the importance of each input data element for predicting spatial information. A numerical evaluation using actual vehicle mobility data demonstrated that a method based on our framework can reduce the uplink traffic volume while achieving the same level of prediction accuracy as the benchmark method.

Furthermore, since different feature selection methods may produce different importance scores for the same data element, we presented two extension schemes that solve that problem in the proposed framework by using the ensemble of importance scores obtained by multiple feature selection methods. These two extension schemes were validated through numerical evaluation.

Prioritization based on data importance can be performed by using a wide variety of means including wireless network selection, base station (access point) selection, channel selection, bandwidth assignment, transmission power control, and media access control. Future work includes to design and evaluate these various means in detail. The issue of privacy risk in mobile crowdsensing (MSC) will also need to be considered. Since MSC relies on data provided by the general public, the privacy issue has been well discussed as detailed by Christin [29]. It is expected that by limiting collected data to only important data for prediction, the proposed framework can easily reduce privacy risk compared with the case where all data are collected.

Abbreviations

MCS: Mobile crowdsensing; SDN: Software-sefined network; IoT: Internet of Things; RF: Random forest; MSE: Mean squared error; MLP: Multi-layer perceptron; NN: Neural network; LSTM: Long short-term memory; GPS: Global positioning system; 3D-CNN: 3D-Convolutional NN; RNN: Recurrent NN; RMSLE: Root Mean Squared Logarithmic Error; RMSE: Root Mean Squared Error

Acknowledgements

We are grateful to Mr. Kota Nakashima and Mr. Takumi Sakai, who are graduate students at Kyoto University, for their help with the data analysis and to Dr. Takehiro Sato, who is an assistant professor at Kyoto University, for his suggestions.

Authors' contributions

The authors propose a framework of data assessment and prioritization that reduces the uplink traffic volume while maintaining the prediction accuracy of spatial information. In the proposed framework, machine learning is used to estimate the importance of each data element and to predict spatial information under the limitation of available data. The authors read and approved the final manuscript.

Funding

This work was supported by the JST PRESTO Grant no. JPMJPR1854 and JSPS KAKENHI Grant no. JP17H01732.

Availability of data and materials

The authors are ready to provide any data obtained through this work upon request.

Competing interests

The authors declare that they have no competing interests.

Received: 3 September 2019 Accepted: 15 April 2020
Published online: 11 May 2020

References

1. Y. Lv, Y. Duan, W. Kang, Z. Li, F. Y. Wang, Traffic flow prediction with big data: a deep learning approach. IEEE Trans. Intell. Transp. Syst. **16**(2), 865–873 (2015)
2. A. Narain, The global GIS and spatial analytics market to touch US$88.3 Billion by 2020 (2018). https://www.geospatialworld.net/blogs/gis-and-spatial-analytics-market. Accessed 24 Aug 2019
3. S. Du, M. Ibrahim, M. Shehata, W. Badawy, Automatic license plate recognition (ALPR): a state-of-the-art review. IEEE Trans. Circuits Syst. Vi. Technol. **23**(2), 311–325 (2013)
4. Image sensor market by technology (CMOS, CCD), processing type (2D, and 3D), spectrum (visible, and non-visible), array type (linear and area), vertical (automotive, consumer electronics, industrial), and geography - global forecast to 2023 (2018). https://www.marketsandmarkets.com/Market-Reports/Image-Sensor-Semiconductor-Market-601.html. Accessed 24 Aug 2019
5. X. Wang, W. Wu, D. Qi, Mobility-aware participant recruitment for vehicle-based mobile crowdsensing. IEEE Trans. Veh. Technol. **67**(5), 4415–4426 (2017)
6. S. Kumar, S. Gollakota, D. Katabi, in *Proc. the first edition of the MCC workshop on mobile cloud computing*, a cloud-assisted design for autonomous driving (ACM, 2012). https://doi.org/10.1145/2342509.2342519
7. M. Gevrey, I. Dimopoulos, S. Lek, Review and comparison of methods to study the contribution of variables in artificial neural network models. Ecol. Model. **160**(3), 249–264 (2003)
8. G. Chandrashekar, F. Sahin, A survey on feature selection methods. Comput. Electr. Eng. **40**(1), 16–28 (2014)
9. R. Shinkuma, T. Nishio, in *Proc. IEEE ICDCS 2019 Workshops - NMIC 2019*, Data assessment and prioritization in mobile networks for real-time prediction of spatial information with machine learning (IEEE, 2019). https://doi.org/10.1109/nmic.2019.00006
10. Y. Chen, T. Farley, N. Ye, QoS requirements of network applications on the Internet. Inf. Knowl. Syst. Manag. **4**(1), 55–76 (2004)
11. A. G. Gotsis, A. S. Lioumpas, A. Alexiou, M2M scheduling over LTE: challenges and new perspectives. IEEE Veh. Technol. Mag. **7**(3), 34–39 (2012)
12. Y. Yamada, R. Shinkuma, T. Sato, E. Oki, Feature-selection based data prioritization in mobile traffic prediction using machine learning. IEEE GLOBECOM CQRM (2018). https://doi.org/10.1109/glocom.2018.8647627
13. Y. Inagaki, R. Shinkuma, T. Sato, E. Oki, Prioritization of mobile IoT data transmission based on data importance extracted from machine learning model. IEEE Access. **7**, 93611–93620 (2019)

14. A. Prinzie, D. Van den Poel, in *Proc. International Conference on Database and Expert Systems Applications*, Random multiclass classification: generalizing random forests to random MNL and random NB (Springer, Berlin, Heidelberg, 2007), pp. 349–358
15. S. Raschka, *Python machine learning*. (Packt Publishing Ltd, Birmingham, 2015)
16. Q. Shuen, R. Diao, P. Su, Feature selection ensemble. Turing-100. **10**, 289–306 (2012)
17. Y. Saeys, T. Abeel, Y. Van de Peer, in *Machine Learning and Knowledge Discovery in Databases*, Robust feature selection using ensemble feature selection techniques (Springer, Berlin, Heidelberg, 2008), pp. 313–325
18. C. B. Samios, M. K. Vernon, in *ACM SIGMETRICS Performance Evaluation Review, vol. 31, no. 1*, Modeling the throughput of TCP Vegas (ACM, 2003), pp. 71–81. https://doi.org/10.1145/885651.781037
19. N. V. Mnisi, O. J. Oyedapo, A. Kurien, in *2008 Third International Conference on Broadband Communications, Information Technology & Biomedical Applications*, Active throughput estimation using RTT of differing ICMP packet sizes (IEEE, Gauteng, 2008), pp. 480–485
20. Y. Ling, J. Chennikara, W. Chen, O. Altintas, Estimator for end-to-end throughput of wireless networks (2009). U.S. Patent No. 7,477,602
21. S. Martello, D. Pisinger, P. Toth, Dynamic programming and strong bounds for the 0-1 knapsack problem. Manag. Sci. **45**(3), 414–424 (1999)
22. S. Ji, W. Xu, M. Yang, K. Yu, 3D Convolutional neural networks for human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 221–231 (2013)
23. Y. Tian, L. Pan, in *Proc. 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, Predicting short-term traffic flow by long short-term memory recurrent neural network (IEEE, Chengdu, 2015), pp. 153–158
24. D. P. Kingma, J. Ba, Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
25. Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, J. Zhang, Edge intelligence: paving the last mile of artificial intelligence with edge computing. Proc. IEEE. **107**(8), 1738–1762 (2019)
26. M. Zeng, T. Yu, X. Wang, V. Su, L. T. Nguyen, O. J. Mengshoel, in *ACM KDD 2016 Workshop on Machine Learning for Large Scale Transportation Systems (LSTS)*, Improving demand prediction in bike sharing system by learning global features (New York, 2016)
27. J. S. Olsson, in *Proc. the 15th ACM international conference on Information and knowledge management*, Combining feature selectors for text classification, (2006), pp. 798–799. https://doi.org/10.1145/1183614.1183736
28. H. Wang, T. M. Khoshgoftaar, A. Napolitano, in *Proc. the 2010 Ninth International Conference on Machine Learning and Applications*, A comparative study of ensemble feature selection techniques for software defect prediction (IEEE, Washington, 2010), pp. 135–140
29. D. Christin, Privacy in mobile participatory sensing: current trends and future challenges. J. Syst. Softw. **116**, 57–68 (2016)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen journal and benefit from:

- ► Convenient online submission
- ► Rigorous peer review
- ► Open access: articles freely available online
- ► High visibility within the field
- ► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com