Doctoral Thesis

Importance-Aware Information Networking toward Smart Cities

Yuichi INAGAKI

Graduate School of Informatics, Kyoto University

September 2021

Preface

The concept of smart cities has been evolving. While the evolving concept of smart cities does not have a fully shared and globally accepted definition, it is possible to describe the most common characteristics of smart cities as implementing the latest technologies to obtain benefits in a wide range of domains. Technological fields such as device networks, sensing, and information analysis enable smart cities to provide various services. As smart city relates to different fields of services, it brings challenges for smart cities, one of which is information networking. As the information networking towards smart cities should satisfy requirements from heterogeneous services, the metrics on which the information networking based should be able to describe requirements from different types of services. However, conventional metrics for information networking are insufficient for smart cities because they are quantized in the network layer, and thus they do not always describe requirements for various services. Therefore, information networking that quantifies the important attributes of requirements of various services in the service layer, which is called "importance-aware information networking" in this thesis, is essential for smart cities. This thesis studies three specific problems on information networking for smart cities based on data importance, each of which focuses on a typical application scenario.

Firstly, this thesis proposes a system for device sharing based on importance extracted from online social relationships between a device owner and user. For smart city services, users provide sensing ability, computation capacity, or network connectivity of their personal devices to smart city services by sharing their devices with other users. When device owners share the limited resources on their devices, they generally want to reduce their costs when they share their devices with someone who is less socially close to them. The proposed system in this work automatically determines how much resources the user can use by acquiring and evaluating online social relationships between a device owner and user as a metric of the importance of transmitted data among devices. This work presents a prototype implementation and a large-scale simulation using a dataset of a real social network. The results show that the proposed system limits resource usage for guest users who are not as close to the device owners. The overhead of the authentication process in the system does not interfere with the resource sharing with guest users close to the device owners.

Secondly, this thesis proposes an Internet of Things (IoT) device control system that uses the importance of data to reduce the amount of transmitted data for input of a machine learning model while maintaining prediction accuracy. Predicting real-time spatial information from data collected by mobile IoT devices is one of the most common structures of smart city services. Mobile IoT devices for real-time spatial information prediction generate an extremely high volume of data, making it impossible to collect all of it through mobile networks. Simply reducing the volume of transmitted data does not ensure the prediction accuracy of real-time spatial information. This work presents an IoT device control system that reduces the amount of transmitted data used as input for real-time prediction while maintaining prediction accuracy. In this work, the proposed system is evaluated with a real-world vehicle mobility dataset in two practical scenarios using the random forest model, an extensively used machine learning model. The results show that the proposed system reduces the amount of transmitted input data for real-time prediction while achieving the same level of prediction accuracy as benchmark methods.

Thirdly, this thesis proposes a framework that periodically updates a machine learning model used to reconstruct the partially collected data by evaluating the importance of the data in terms of both inference and re-training and prioritizing collecting important data. Sparse mobile crowdsensing is a crowdsensing paradigm that reduces the sensing cost while ensuring data quality by collecting data sparsely and reconstructing desired data using inference algorithms, including machine learning algorithms. However, real-time inference of spatial information with sparse mobile crowdsensing has not sufficiently considered the change of the nature of data over time. As a result, the accuracy of the reconstructed data can deteriorate over time. This work presents a framework that periodically updates a machine learning model used for reconstructing data by evaluating the importance of the data in terms of both inference and re-training and prioritizing collecting important data. The evaluation results show that the proposed system with periodical model updates performed better in accuracy than the benchmarks over time.

This thesis is organized as follows. Chapter 1 introduces the background of importance-aware information networking towards smart cities. Chapter 2 introduces related technologies to this work. Chapter 3 develops device sharing based on social importance. Chapter 4 introduces prioritized transmission of sensing data based on data importance for inference by machine learning model. Chapter 5 presents a periodic update of a machine learning model for sensing data analysis based on data importance for learning. Finally, Chapter 6 concludes this thesis.

Acknowledgements

This thesis is the summary of my doctoral study at Kyoto University, Kyoto, Japan. I am grateful to a large number of people who have helped me to accomplish this work.

First of all, I would like to express my sincere gratitude to my supervisor, Professor Eiji Oki, for his continuous support and guidance toward my research. Without his persistent help and constant encouragement, I would have never completed this work.

I owe my deepest gratitude to Professor Ryoichi Shinkuma of the Faculty of Engineering, Shibaura Institute of Technology, for giving me the opportunities to do research and providing invaluable guidance throughout this research. His dynamism, vision, sincerity, and motivation have deeply inspired me to accomplish this work. I would like to express my appreciation to Assistant Professor Takehiro Sato of the Graduate School of Informatics, Kyoto University, for his comments and helps on my papers.

I would like to express my deep appreciation to Professor Hiroshi Harada and Professor Sadao Kurohashi of the Graduate School of Informatics, Kyoto University, for valuable advice and incisive comments. Their advice and comments have been a great help in improving this thesis.

I am extremely grateful to Ms. Mariko Tatsumi and all members of the Oki Laboratory of the Graduate School of Informatics, Kyoto University, for their kind helps in my research and life.

Finally, I would like to thank my parents, family, and Ms. Yu Ito. Without their understanding, constant support, and encouragement, I would have never accomplished this work.

Contents

List of Figures

List of Tables

Notations

Notation	Description
$E(x, y)$	One-to-one relationship between nodes x and y
$\Gamma(z)$	Set of neighbors of a node z
k_z	Degree of a node z
$\text{paths}_{xy}^{\langle l \rangle}$	Set of all paths with length l connecting nodes x and y
$rank(A)$	Rank of a matrix A
$\|\|A\|\|_F$	Frobenius norm of a matrix A
A^T	Transpose of a matrix A
$A \circ B$	Element-wise product of matrices A and B

Abbreviations

Abbreviation	Description
ISO	international organization for standardization
IEC	international electrotechnical commission
ITU	international telecommunication union
IEEE	institute of electrical and electronics engineers
U4SSC	united for smart sustainable cities
ICT	information and communication technologies
KPI	key performance indicator
SSC	smart sustainable city
QoS	quality of service
IoT	Internet of Things
SNS	social networking service
PCA	principal component analysis
MI	mutual information
GA	genetic algorithm
PSO	particle swarm optimization
MLP	multilayer perceptron
LSTM	long short-term memory
WSN	wireless sensor network
DTN	delay-tolerant network
LTE	long-term evolution
LTE-A	LTE-Advanced
BS	base station
PC	personal computer
DB	database

Abbreviation	Description
OSN	online social network
URL	uniform resource locator
AP	access point
API	application programming interface
HTTP	hypertext transfer protocol
PaaS	platform as a service
CDF	cumulative distribution function
WHO	world health organization
GDP	gross domestic product
UAV	unmanned aerial vehicle
CPS	cyber-physical system
LiDAR	light detection and ranging
RMSE	root mean square error
RMSLE	root mean squared log error
KNN	K-nearest neighbors
CNN	convolutional neural network
SVD	singular value decomposition
KNN-S	spatial KNN
KNN-T	temporal KNN
QBC	query by committee

Chapter 1

Introduction

1.1 Background

1.1.1 Concept of smart cities

The concept of a smart city has become popular worldwide over the past several years. One of the most common characteristics of a smart city is focusing on providing social benefits, economic growth, and creating new opportunities by adopting the latest technology for the benefit of a wide range of domains [1]. A smart city provides various services for various domains employing technologies from many different scientific fields.

Several definitions of a smart city have been defined at different times by various organizations and stakeholders. There is no unified definition of a smart city because a smart city involves various domains and provides various services. In 1993, the first idea of a smart city was presented by Singapore city when Singapore city announced itself as an “intelligent city” [2]. Between 2000 and 2010, the concept of ”digital city” appeared. The concept of a digital city is closely related to a smart city. One study defined a digital city as an open, complex, and adaptive system based on computer networks and urban information resources that form a virtual digital space for the city [3]. Another study defined a digital city as one that collects and organizes the digital information of the corresponding city and provides a public information space for the people living in and visiting it [4]. The digital city can be considered as a precedent for future ideas of a smart city. In 2007, one of the first ideas of the term smart city was presented, which defined a smart city as a city that is well-performing in a forward-looking way in the industry, education, citizen participation, and technical infrastructure fields [5]. After 2010, as smart cities begin to gain interest rapidly, the number of definitions has been increasing significantly. One known definition was published by the IBM company. IBM defined a smart city as an instrumented, interconnected, and intelligent city [6]. In other words, a smart city collects real-world data from both physical and virtual sensors in real-time, aggregates and shares those collected data among the various city services, and performs practical analysis and visualization to improve operational business processes. Another known definition presents a smart city as a sustainable and efficient city that functions by integrating all infrastructures and services and using intelligent devices to monitor and control the city [7].

Several international organizations have developed international standards to establish definitions and methodologies for the research and development of smart cities. The following organizations are involved in such standardization: the international organization for standardization (ISO), the international electrotechnical commission (IEC), the international telecommunication union (ITU), and the institute of electrical and electronics engineers (IEEE). ISO 37122:2019 (Sustainable Cities and Communities – Indicators for Smart Cities) specifies and establishes definitions and methodologies in various domains to provide a complete set of indicators to measure progress towards a smart city [8]. IEC has identified over 1800 standards that already impact smart cities [9]. The SyC Smart City committee in IEC is actively promoting the development of standards to assist in operating smart cities. The SyC Smart City committee is currently developing a new standard, IEC 63152, which provides city planners with a best practice tool to protect a variety of city services from destruction caused by disasters [10]. ITU established groups such as Study Group 20 and the united for smart sustainable cities (U4SSC) initiative to encourage the use of information and communication technologies (ICTs) and to develop a set of international key performance indicators (KPIs) for smart sustainable cities (SSCs) [11]. IEEE established the IEEE Smart Cities Community to bring together IEEE's broad range of technical societies and organizations to promote state-of-the-art smart city technologies [12]. In 2017, IEEE P2784 (Smart City Planning Guide) was published [13]. This standard includes a framework that provides a methodology for planning innovative, scalable, and sustainable solutions for smart cities.

1.1.2 Information networking for smart cities

As smart city relates to many fields of services, it brings challenges for smart cities, one of which is information networking. As various services have been provided by smart cities, networks for smart cities need to meet various requirements and challenges of various services. One of the most common conventional approaches for information networking is to use metrics based on the quality of service (QoS) [14, 15]. Building QoS-based information networking architecture consists of three steps; analysis, quantification, and mapping [16]. First, in the analysis step, service requirements are analyzed in the service layer, where the QoS category will be extracted as a result of the analysis. Second, in the quantification step, attributes of service requirements are quantified in the network layer according to the QoS category. Third, in the mapping step, the QoS requirements are mapped to the network attributes so that the network will select the proper mechanism for networking. For example, for a mobile streaming service where videos gathered by mobile cameras are transferred to a remote server through a mobile network, the three steps are as follows. First, continuous transmission of large data is a requirement for the service. Second, acceptable bandwidth, latency, and jitter are quantified from the service requirement. Third, these metrics are mapped to a packet transmission strategy.

1.2 Issue of information networking for smart cities

As the information networking towards smart cities should satisfy various requirements from various services, the metrics on which the information networking based should be able to describe various requirements for various services.

However, conventional QoS-based information networking is insufficient for smart cities because the quantification of QoS attributes in the network layer does not always describe various requirements for various services. Users of networks for smart city services can be roughly categorized into two; humans and machines. For humans, social activities such as sharing their data or resources with others have become a fundamental aspect of services [17–20]. For these social activities, the importance of social relationships among the users is a key metric. On the other hand, for machines, the efficiency of communication and accuracy of results of data analysis is one of the major requirements [21–23]. To satisfy these requirements, how important a part of transmitted data is for a service is a key metric. Quantifications to obtain these metrics can be performed in the service layer but not in the network layer.

Therefore, information networking that quantifies the important attributes of requirements of various services in the service layer, which is called "importance-aware information networking" in this thesis, has become an essential requirement for smart cities.

1.3 Problem statements

This thesis studies three specific problems about importance-aware information networking for smart cities, each of which includes a question that has not been addressed and is answered in this thesis.

1.3.1 Device sharing based on social importance

Device network is one of the key technological factors for smart cities. Device sharing is one of the applications that are provided on top of a device network. When people share devices, they would be concerned about costs such as battery or bandwidth. In addition, device owners generally want to reduce their costs when they share their devices with someone who is less socially close to them [24–28]. Device sharing systems need to meet demands in which device owners want to restrict less socially close users from using the resources of the owners' devices.

A question arises: *is there any system for device sharing that respects social relationships among device owners and users?* This question has not been addressed, and is studied in Chapter 3.

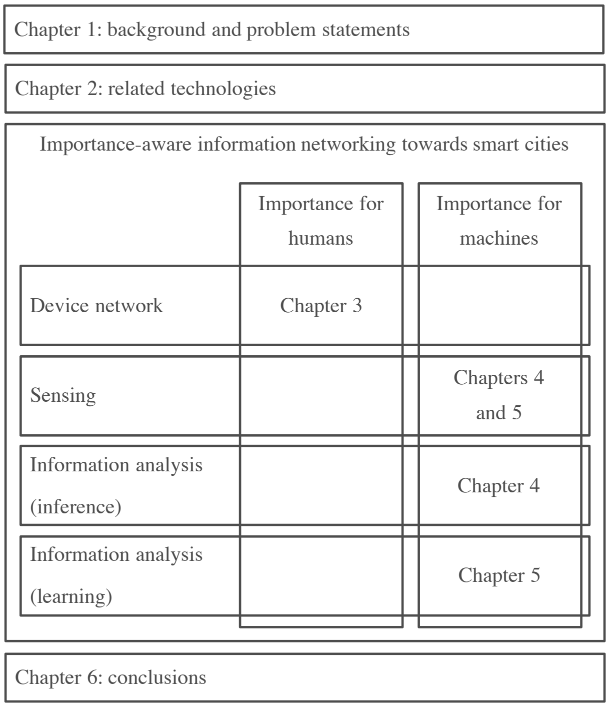
1.3.2 Prioritized transmission of sensing data based on data importance for inference by machine learning model

Sensing and information analysis of data collected by sensing are key enablers of smart cities. Advances in technologies such as wireless communications and mobile Internet of Things (IoT) sensors made it possible to obtain sensor data almost everywhere and at any time. The collected sensor data is uploaded to servers and processed using machine learning models to infer valuable information for smart city services. Mobile IoT sensors generate an extremely high volume of data, making it impossible to collect all of it through mobile networks. Although reducing the total volume of transmitted data is essential, reducing the data transmitted to servers may lead to poor inference accuracy.

A question arises: *is there any system that reduces the amount of transmitted sensor data used as input for inference while maintaining the inference accuracy by transmitting only an important part of sensor data that contributes to the inference accuracy?* This question has not been addressed, and is studied in Chapter 4.

1.3.3 Periodic update of machine learning model for sensing data analysis based on data importance for learning

Sensing combined with information analysis of sensed data is one of the most common architecture for smart city services. Sensors collect data and transmit the collected data to the server, and then the server aggregates the collected data and performs inference on the collected data. As the nature of data changes over time, the accuracy of the inference results can deteriorate over time unless the inference model is re-trained using the newly collected training data. The data for re-training, as well as the data for inference, consumes



Chapter 1: background and problem statements

Chapter 2: related technologies

Importance-aware information networking towards smart cities

	Importance for humans	Importance for machines
Device network	Chapter 3	
Sensing		Chapters 4 and 5
Information analysis (inference)		Chapter 4
Information analysis (learning)		Chapter 5

Chapter 6: conclusions

Figure 1.1: Chapter overview of this thesis.

sensing costs. The problem stated in Section 1.3.2 only considers how to collect the data for inference efficiently while maintaining the inference accuracy, which means it does not consider how to collect the data for re-training.

A question arises: *is there any framework that efficiently collects both the data important for re-training the inference model and the data important for inference in order to maintain long term accuracy of the inference?* This question has not been addressed, and is studied in Chapter 5.

1.4 Overview and contributions of this thesis

Figure 1.1 shows the chapter overview of this thesis. Chapter 2 introduces related technologies to this work.

Chapter 3 proposes a system for device sharing based on importance extracted from online social relationships between a device owner and user. The proposed system in this work automatically determines how much resources the user is allowed to use by acquiring and evaluating online social relationships between a device owner and user as a metric of the importance of transmitted data among devices. This work presents a prototype implementation and a large-scale simulation using a dataset of a real social network. The results show that the proposed system limits resource usage for guest users who are not as close to the device owners. The overhead of the authentication process in the system does not interfere with the resource sharing with guest users close to the device owners.

Chapter 4 proposes an IoT device control system that uses the importance of data to reduce the amount of transmitted data for input of a machine learning model while maintaining the prediction accuracy. This work presents an IoT device control system that reduces the amount of transmitted data used as input for real-time prediction while maintaining prediction accuracy. In this work, the proposed system is evaluated with a real-world vehicle mobility dataset in two practical scenarios using the random forest model, an extensively used machine learning model. The results show that the proposed system reduces the amount of transmitted input data for real-time prediction while achieving the same level of prediction accuracy as benchmark methods.

Chapter 5 proposes a framework that periodically updates a machine learning model used for reconstructing the partially collected sensor data by evaluating the importance of the data in terms of both inference and re-training and giving priority to collecting important data. This work presents a framework that periodically updates a machine learning model used to reconstruct data by evaluating the importance of the data in terms of both inference and re-training and prioritizing collecting important data. The evaluation results show that the proposed system with periodical model updates performed better in accuracy than the benchmarks over time.

Finally, Chapter 6 concludes this thesis and discusses the future works to extend this work.

Chapter 2

Related technologies

2.1 Importance estimation from network graph models

This section presents several common metrics that help us estimate network graphs.

Social relationships among people are one of the most common examples of a network graph. Communities on most social networking services (SNSs) can be explicitly created by users. For example, such communities are called “groups” on Facebook. However, communities can be detected from the network topology by using community-detection algorithms. Link communities [29] detect communities that users belong to by hierarchically clustering the links between users. The most remarkable feature of this algorithm is that it allows users to belong to multiple communities.

One-to-one relationships between two nodes can also be used to analyze social relationships. The one-to-one relationship between nodes x and y can be represented by $E(x, y)$.

The $E(x, y)$ in common neighbors [30] is given as

$$E(x, y) = |\Gamma(x) \cap \Gamma(y)|, \tag{2.1}$$

where $\Gamma(z)$ is the set of neighbors of a node z. It is assumed that two users who share many common neighbors are likely to have a stronger relationship.

The $E(x, y)$ in the Jaccard Index [31] is given as

$$E(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}. \tag{2.2}$$

It is assumed that two users have a stronger relationship when the set of their common neighbors matches well.

The $E(x, y)$ in the Adamic-Adar Index [32] is given as

$$E(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log k_z}, \tag{2.3}$$

where k_z is the degree of a node z. It formalizes the intuitive notion that rare features are more important.

The $E(x, y)$ in the Katz Index [33] is given as

$$\begin{aligned} E(x, y) &= \sum_{l=1}^{\infty} \beta^l \cdot |\text{paths}_{xy}^{\langle l \rangle}| && (2.4) \\ &= \beta A_{xy} + \beta^2 (A^2)_{xy} + \beta^3 (A^3)_{xy} + \cdots, && (2.5) \end{aligned}$$

where $\text{paths}_{xy}^{\langle l \rangle}$ is the set of all paths with length l connecting x and y, β is a free parameter controlling the path weights, and A is the adjacency matrix: $A_{xy} = 1$ if x and y are directly connected and $A_{xy} = 0$ otherwise. Note that, $(A^l)_{xy}$ is equal to the number of paths of length l from x to y. It gives the shorter paths greater weight.

2.2 Importance estimation from machine learning models

2.2.1 Feature selection

Feature selection was originally considered as a method for selecting a set of variables (features) from the input that can efficiently describe the input data while reducing effects from noise or irrelevant variables and still provide good prediction results [34]. Feature selection methods can reduce computation time, improve prediction performance, and provide a better understanding of the data in machine learning or pattern recognition applications. Feature selection differs from other dimension reduction methods such as principal component analysis (PCA) in that it does not create new features since it uses the input features themselves to reduce their number.

Existing methods of feature selection can be roughly categorized into three: filter methods, wrapper methods, and embedded methods [34]. Each group of methods is described and characterized as follows.

Filter methods use variable ranking techniques as the primary criterion for variable selection by ordering. Filter methods act as preprocessing to rank the features in advance before machine learning. Filter methods measure feature relevance to filter out the less relevant variables. Feature relevance is a metric of the feature's usefulness in describing data. There are two criteria for understanding the relevance of a feature. Pearson correlation coefficient is one of the simplest criteria for filter methods [35]. Ranking based on correlation is simple but can only detect linear dependencies between input variable and output. Another criterion is mutual information (MI), which is used to measure dependency between two input variables in information theory [36]. This method is also simple but can result in poor performance because inter-feature MI is not considered [37]. The advantages of filter methods are that they are computationally light and avoid overfitting since they do not rely on learning algorithms. On the other hand, in filter methods, features that are less informative on their own could be discarded even if they are informative when combined with other features.

Wrapper methods use the predictor as a black box and the prediction accuracy as the objective function to evaluate the subset of variables. Since exhaustive search methods for all subsets of variables in large datasets requires enormous computational resources, simplified search algorithms are employed to find suboptimal subsets. Two search algorithms are used for wrapper methods: sequential selection algorithms and heuristic search algorithms. The sequential selection algorithms start with an empty or complete set of variables and repeatedly add or remove variables until the objective function is maximized [38]. The heuristic search algorithms such as genetic algorithm (GA) [39] or particle swarm optimization (PSO) [40] are used to find a local optimum for prediction accuracy as the objective function. The main disadvantage of the wrapper method is the large number of computations required to obtain the resulting feature subset. The predictor needs to be trained and evaluated iteratively with each subset of features. Another drawback is that the predictors are prone to overfitting [41]. When prediction accuracy is used for feature selection, the output feature subset may have high accuracy for known data but only low generalization ability for unknown data.

Embedded methods [35, 42] perform feature selection in the training process. Compared to wrapper methods, embedded methods reduce the computation time taken for repeated training and evaluation for different subsets in wrapper methods. Weights method and impurity method are the most common among embedded methods. Weights method calculates the feature importance from the trained neural network model using a saliency measure. Weights method can be applied to neural-network-based models including a multilayer perceptron (MLP) and a long short-term memory (LSTM) model. The impurity method calculates the feature importance from a metric called "impurity", which is used during the training of decision tree models. At each node of the decision tree, m elements are randomly selected out of all the features, and the division method that best divides the m elements is selected. At each node t of the binary tree, the optimal splitting is searched for using impurity $i(t)$ [43]. Impurity is a measure of how well the potential division at that node divides the sample into two classes. This means that the impurity reflects the importance of the elements used in the splittings. The impurity method can be applied to decision-tree-based models, including the random forest model [44].

There are feature selection methods that do not belong to any of the above three categories but are commonly used. The perturb method evaluates the effect on the output when the value of each input is slightly changed in machine learning models [45]. The method changes the input values of one variable and measures the effect on the output. The accuracy of the output deteriorates more when the changed variable is more important. The perturb method is straightforward and applicable to a wide variety of machine learning models. Another common method is feature selection ensemble. Feature selection ensemble is an ensemble-based method that aims to construct a group of feature subsets and then produce an aggregated result from the group [46]. Feature selection ensemble improves the robustness of feature selection because the performance variance can be reduced compared to that of a single result obtained from a single approach.

2.2.2 Importance estimation using prediction errors

This section introduces methods that calculate weights or perform data sampling using the error between the real values and the predicted values by machine learning models.

Generalization error can be used for sampling the training data for neural network models [47]. This method filters out a larger amount of data if the data has a lower generalization error. Low generalization error means that the model already performs well for that data. Thus that data can be considered to have a small contribution as training data for updating the model.

Boosting, which is a popular technique of machine learning, uses prediction error for weighting each data [48]. Boosting is an ensemble learning method that aggregates multiple results from multiple prediction models to create a better model [49]. When the models are trained, one prediction model uses training data which is weighted by the prediction error of another prediction model. The prediction models trained by boosting method can compensate for weak points of each other and the prediction accuracy can be improved.

Chapter 3

Shared-resource management using online social relationship metric for altruistic device sharing

3.1 Overview

Over the past several years, we have witnessed great progress in wireless communications and digital electronics. These advances have enabled an increasing number of devices, such as tablets, sensors, wearable devices, robots, and autonomous cars, to be connected to the Internet. Due to the spread of the IoT paradigm, even everyday items, such as food packaging, furniture, and paper documents, will be Internet nodes by 2025 [50]. In addition to this change, a global trend toward peer-to-peer sharing of personal assets has been suggested. This trend is called the "sharing economy" and is demonstrated in services such as Airbnb, Uber, and Freecycle. The sharing economy was nominated by Time in 2011 as one of "10 ideas that will change the world" [51]. Furthermore, the global annual revenue of the sharing economy, which was \$15 billion in 2015, has been estimated to grow to \$335 billion by 2025 [52].

Due to the confluence of the above two paradigms, i.e., IoT and sharing economy, various devices owned by a person will be shared with others. For example, members of a global WiFi sharing community called FON [53] share their WiFi routers with other members. Another example is mobile cloud [54] [55]. By sharing computing resources with mobile devices, mobile cloud attains more powerful computing than stand-alone computing and enables mobile devices to offload computing tasks with low levels of latency. Sensing devices in wireless sensor networks (WSNs) are also shared for various purposes. SenseWeb is an infrastructure for shared sensing, which provides greater understanding by collecting sensing data from multiple different networks [56]. Sharing airborne sensors enables efficient use of their spare sensing resources [57] [58]. A system called eShare enables energy exchange among shared sensors [59].

When a device owner decides how much or how long her or his device can be shared with others, it is a good idea to consider how close these others are to the device owner. There are two reasons for this. First, social closeness has a strong relationship with our daily mobility patterns. We have more chance to encounter someone if she or he is socially close to us. Eagle et al. introduced a system to collect data from mobile phones and studied the relation between the logged data and social nature of the subjects [60]. They revealed that social closeness between people is strongly correlated with their contact logs. Hui et al. presented a delay-tolerant network (DTN) based on social metrics [61]. To infer human communities and select forwarding paths, they measured the social closeness between two people by the number of contacts and how long they spend together. Second, the social closeness between people has a correlation with how altruistic someone will be to others [24] [25] [26] [27] [28]. For example, when devices are shared among people, the owners do not want to share their devices with strangers, while they are more willing to share their devices with their socially closer friends or families. The less socially close the guest user is to the device owner, the less altruistic the device owner becomes.

Device-sharing systems need to meet demands in which device owners want to restrict less socially close users from using the resources of the owners' devices. A typical example of altruistic device sharing, which this work will focus on in Section 3.3, is 'tethering' in cellular networks: an owner who uses a personal device such as a smartphone, which has direct connectivity to cellular networks such as 3G, long-term evolution (LTE), or LTE-Advanced (LTE-A), relays data from/to base stations (BSs) for others who do not have direct connectivity to cellular networks but connect their personal computers (PCs) or tablets to the owner's device via WiFi [62]. Tethering incurs costs such as those imposed by battery life or bandwidth [63]. When device owners offer tethering, they want to save the costs to guest users who are not as close to them because they are less altruistic to such users. However, conventional device-sharing services do not meet such demand. They do not allow device owners to vary the authorized level of resource usage of guest users or only allow device owners to manually manage the authorized level of resource usage of users, which imposes a great burden on device owners.

This chapter proposes a system that uses online social relationships to meet device owners' demands for resource management to enable altruistic device sharing. When a shared device receives a connection request from a guest user, the shared device first sends a request to the authentication server. Then, the authentication server evaluates online social relationships and determines how much of a resource on the shared device can be used by a guest user. This work also presents a prototype implementation and a large-scale simulation using a dataset of a real social network to verify that i) the proposed system limits the resource usage for guest users who are not as close to the device owners, and ii) the overhead of the authentication process in the system does not interfere with the resource sharing with guest users who are close to the device owners.

Several studies have been carried out that are similar to this work. Shankar et al. presented and demonstrated an architecture called SBone, which allows personal devices to seamlessly and securely share their resources and state with each other by using a social network for authentication, naming, discovery, and access control [64]. They suggested that SBone would be applicable to situations in which a device owner provided her or his Internet connectivity to others who were friends with her or him in online social networks. Another similar effort has been in communication with social-aware device-to-device, which directly share data between mobile devices used by people who have social relationships without using infrastructure networks such as cellular networks [65] [66] [67]. However, these prior studies did not consider how shared resources were to be managed on the basis of social closeness between owners and users.

The rest of this chapter is organized as follows. Section 3.2 presents the architecture and resource management procedures for the proposed system. Section 3.3 provides a prototype implementation and simulation results that validate the performance and effectiveness of the proposed system. Finally, Section 3.4 concludes this chapter.

3.2 Proposed system design

3.2.1 System architecture

The proposed system architecture (Fig. 3.1) consists of four components: (a) an authentication server, (b) shared devices, (c) owners, and (d) guest users. The authentication server manages the shared devices and the online social account information of the owners and guest users. The authentication server determines which guest user can access which function or resource of the shared devices according to the relationship between the owners and guest users. A centralized architecture is adopted for the authentication server, so it can easily manage online social relationships between the owners and guest users. The shared devices are devices that can be accessed by guest users, such as tablets, sensors, wearable devices, robots, and autonomous cars. Each shared device belongs to one owner. The guest users are granted access to the shared devices according to the online social relationship with the owner of the shared devices.

3.2.2 Owner-related procedures

Device registration

An owner registers her or his personal devices on the authentication server before the owner starts to share the devices. When an owner registers a device, the authentication server issues a unique ID to the device. The authentication server associates the device ID with the owner's online social account information and records them in a database (DB).

Social-closeness evaluation from extracted social relationships

The proposed system requires i) a data source from which the proposed system obtains online social relationships and ii) a metric by which the proposed system quantitatively analyzes the online social relationships to use those relationships between owners and guest users to manage resources.

One of the most common and familiar examples of online social relationships is found in online social networks (OSNs) [68]. OSNs are offered by SNSs such as Facebook, Twitter, Google+, and LinkedIn. OSNs consist of nodes and edges. Nodes represent users (more specifically, online social accounts of users) of OSNs, while edges represent social interactions among these users. Note, in this section, users mean not device users but SNS users. The most basic social interactions that are represented by edges are friendships. Although some OSNs adopt undirected friendships and others adopt directed friendships, both types of friendships are included in online social relationships. Comments, messages, and reactions to other users are also examples of online social relationships, apart from friendships.

Several common metrics can be used to analyze the social closeness between users, as described in Section 2.1. By using communities and one-to-one relationships between two users, the proposed system defines the social closeness between x and y as

$$SC(x, y) = \begin{cases} 0 & \text{if } x \text{ and } y \text{ are not friends, or} \\ & \text{they do not belong to the same} \\ & \text{community} \\ E(x, y) & \text{otherwise} \end{cases}, \tag{3.1}$$

where $E(x, y)$ is an index that represents the one-to-one relationships between x and y, as defined in Section 2.1.

3.2.3 User-related procedures

The authentication flow of the proposed system is illustrated in Fig. 3.2. Details of each message in Fig. 3.2 are described in Table 3.1. Authentication consists of two phases: identification and authorization. The authentication server identifies guest users in the identification phase (1.1–1.4) by using their online social accounts. The authentication server acquires the online social relationships between the owner and guest user, then the shared devices control the access for the guest user based on the relationships in the authorization phase (2.1–2.4).

Identification

(1.1) A guest user requests access to the shared device. (1.2) The shared device requests the guest user to sign in to the authentication server. (1.3) The guest user signs in to the authentication server with the guest user's online social account. (1.4) The authentication server notifies the shared device that the guest user has completed signing in to the authentication server.

Authorization

(2.1) The shared device requests the authentication server to authorize the guest user. (2.2) The authentication server acquires online social relationships between the owner and guest user. The authentication server creates access control information based on these relationships that define whether the guest user can access the shared device and the authorized level of resource usage for the guest user. (2.3) The authentication server issues the resource-management information to the shared device. (2.4) The shared device controls access for the guest user based on the received information.

3.2.4 Advantages and disadvantages

With the proper use of online social relationships, services that meet the demand of smart cities can be developed. By combining information acquired from social relationships with free WiFi and business support, the proposed system can be extended to a smart city product. For example, Bumbee Labs in Sweden has offered free WiFi to tourists to acquire their mobility logs and analyze them to increase B to B sales [69]. Combining online social relationships with those data will help such services offer more valuable and interesting analysis.

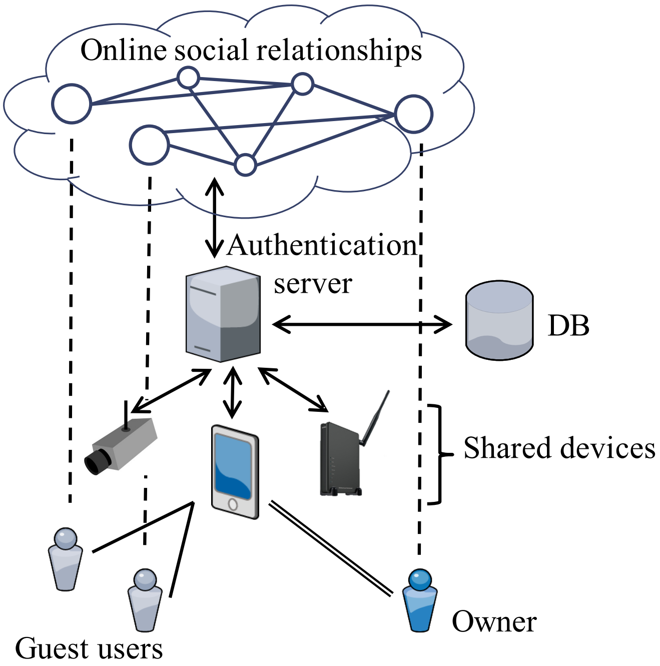


Figure 3.1: Proposed system architecture. (©2018 IEEE.)

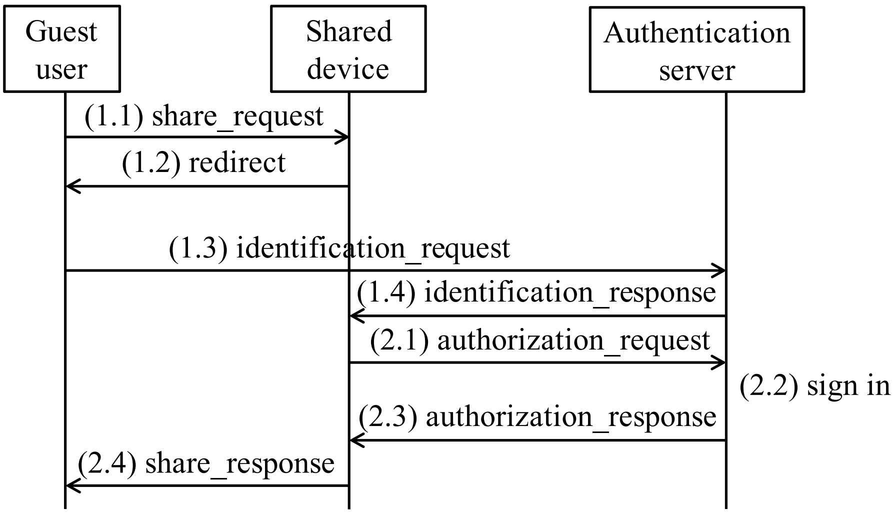

Guest user
Shared device
Authentication server
(1.1) share_request
(1.2) redirect
(1.3) identification_request
(1.4) identification_response
(2.1) authorization_request
(2.2) sign in
(2.3) authorization_response
(2.4) share_response

Figure 3.2: Authentication flow. (©2018 IEEE.)

Table 3.1: Details of exchanged messages.

Message name	Content data name	Details
(1.1) share request	guest_id	ID to identify guest user
(1.2) redirect	authentication_url	Uniform resource locator (URL) of authentication server's endpoint
(1.3) identification request	guest_id	See above
	shared_device_id	ID to identify shared device
	social_account_id	ID of a social account of guest user
	social_account_pass	password of social account of guest user
(1.4) identification response	access_token	Secret key to access online social relationships
(2.1) authorization request	access_token	See above
	owner_id	ID to identify owner
	guest_id	See above
(2.3) authentication response	resource_management_info	Information to control access from guest users e.g., authorized connection time
(2.4) share response	resource_management_info	See above

However, it should also be noted that online social relationships may lead to privacy issues. A major concern is that one user may be able to infer some private information of another user. As future work, the problem of how the social relationships are prone to raise such a risk will be investigated.

3.3 Performance evaluation

In this performance evaluation, a tethering scenario is assumed, in which a device owner relays data to cellular networks, such as LTE, for other guest users who connect their PCs or tablets to the owner's mobile device, such as a smartphone, via WiFi [62]. Section 3.3.1 introduces an implementation of a prototype system and the performance measurement of the prototype system to confirm that the authentication overhead is within a realistic range. Using the authentication overhead actually measured (Section 3.3.2) presents a simulation with large scale and real social network data to verify i) and ii) mentioned in Section 3.1.

For the rest of this section, authorized connection time is used as an index of the authorized level of resource usage. The authorized connection time is the duration in which guest users are permitted to connect to shared devices.

3.3.1 Prototype implementation

Overview

The architecture of the implemented prototype system is illustrated in Fig. 3.3. This prototype system selects the WiFi access point (AP) as a shared device and uses the number of common neighbors on Facebook as an indicator of social closeness. The number of common neighbors [30] is used as a metric, as described in Section 3.2.2 to control the authorized connection time for guest users to access the Internet through the AP. To delegate guest-user identification management to Facebook accounts, the OAuth protocol is used. In addition, the implemented prototype system adopts a system called PacketFence to control the packet flow through the AP. PacketFence communicates with the authentication server and guest device and performs access control on behalf of the shared WiFi AP.

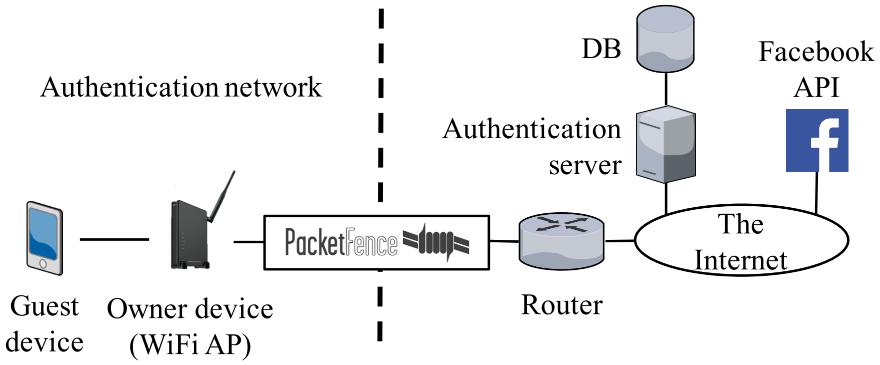


Figure 3.3: Implemented prototype system. (©2018 IEEE.)

The authentication flow is composed of the identification and authorization phases. In the identification phase, the guest user requests access to the shared device and signs in to the authentication server with the guest user's Facebook account. The authentication server identifies the guest user by receiving the guest user's information from Facebook. The authentication server and PacketFence communicate with each other to exchange the guest user's pieces of information such as the guest user's name or email address. In the authorization phase, the authentication server obtains the number of common friends between the owner and guest user and determines the authorized connection time for the guest user to access the WiFi AP.

Under this configuration, the implemented prototype system allows the guest users to connect to the Internet through the AP without entering complex WiFi passwords as long as they have a Facebook account.

Details

Facebook API Facebook offers one of the largest OSNs in the world [70] and offers rich application programming interfaces (APIs). Facebook APIs allow the implemented prototype system to use various data on Facebook easily. The number of common neighbors is an example of various data offered by Facebook through the APIs. These indicators represent the social closeness among users well; therefore, they are suitable for controlling the authorized connection time for each user.

OAuth OAuth is a protocol that enables a third-party application to access resources on a hypertext transfer protocol (HTTP) service on behalf of a resource owner [71]. The OAuth protocol flow consists of the following three main parts. (1) The resource owner is identified by the HTTP service and approves the third-party application's access to the resource. (2) The third-party application receives an access token from the authorization server of the HTTP service. (3) The third-party application requests the protected resource on the resource server of the HTTP service by presenting the access token.

In the implemented prototype system, Facebook, online social relationships on Facebook, and the authentication server of the implemented prototype system represent the HTTP service, resource, and third-party application, respectively. By using the OAuth protocol, the implemented prototype system gains two benefits. First, the implemented prototype system can delegate the identification of users to Facebook. This saves the system the trouble of managing passwords or user accounts on its own. Second, the implemented prototype system can acquire online social relationships from Facebook for access control on behalf of the users.

PacketFence The packet flow through the AP is controlled by a system called PacketFence, which is a free and open source network access control solution [72] that can be deployed under the following three types of enforcement: inline, out-of-band, and hybrid. The implemented prototype system adopts inline enforcement, which is the most basic and simple enforcement among the three. Under inline enforcement, the PacketFence server is placed between a router connected to the Internet and an authentication network that includes the shared AP and guest user devices. Therefore, all packets exchanged between the authentication network and Internet must go through the PacketFence server. When a packet from an authorized guest user device attempts to go through the PacketFence server to outside the authentication network, the PacketFence server behaves like a normal router and allows the packet to pass. On the other hand, when a packet from an unauthorized guest user device attempts to do the same thing, the PacketFence blocks the packet and displays a captive portal that prompts the guest user to sign in.

The flexible design of PacketFence allows the implemented prototype system to add a module to exchange authentication information with the authentication server.

Performance measurement

Metric This section adopts the time required for authentication as a metric of authentication overhead. However, the time consumed while the user enters her or his username and password on the sign-in page of Facebook should not be included in the measurement because it varies from person to person. Therefore, it is assumed that the user usually uses Facebook with a browser on the user's device, i.e., the user has already signed in to Facebook and a Facebook credential has been stored in a browser cookie. Under this assumption, the sign-in procedure is completed as soon as the user visits the sign-in page of Facebook, and the time taken to enter the username and password is not included in the measurement.

Experimental setup The details of the experimental setup are listed in Table 3.2. PacketFence was installed on a CentOS machine. The authentication server was implemented as a Ruby on Rails web server and deployed on one of the most popular platform as a service (PaaS) called Heroku.

The time required for authentication was extracted from timestamps in a log file of the authentication server. In this measurement, the time required for authentication is defined as the length of a period that begins with the first request to the server and ends with the last response from the server.

Reference setup The reference system does not take into account the online social relationships between a device owner and guest users. The authentication server in the reference system does not acquire and evaluate online social relationships on Facebook and allows all guest users to use the WiFi AP for a fixed duration.

Results Table 3.3 shows the duration required for authentication, which was measured using the prototype system. In the table, the 5th shortest, median, and 20th shortest values obtained from 25 measurements are shown for evaluating the distribution of the measured duration. The median of the duration

Table 3.2: Details of experimental setup.

OS	CentOS 6.8
Memory	8 GB
CPU	Core i7-860 2.8 GHz × 8
No. of measurements	25
PacketFence version	6.3.0
Guest device	iPhone 7 iOS 11.2.2
Browser on guest device	Google Chrome
Authentication server	Ruby 2.3.1, Rails 4.2.7, on Heroku

Table 3.3: Time required for authentication.

	5th (s)	median (s)	20th (s)
Reference	4.786	5.086	6.216
Proposed	5.196	5.408	5.800

required for authentication in the proposed system was slightly longer than that in the reference system. This is because the proposed system acquires and evaluates online social relationships on Facebook, while the reference system does not. However, this duration was not dominant in the entire authentication process. These results verified that the proposed system works sufficiently in terms of the overhead for authentication compared with the reference system.

3.3.2 Simulation with real data

Evaluation scenario

In the previous section, measuring the authentication overhead is discussed. In this section, using the measured overhead, a simulation is conducted to verify i) and ii) mentioned in Section 3.1. In the simulation, each user is assumed to have a tethering device and move around cities based on the check-in data of an actual location-based social network.

Figure 3.4 illustrates the evaluation scenario. The simulation takes into account tethering in cellular networks: guest users who are not directly connected to cellular networks send/receive data via a device owner's smartphone. The system in the simulation determines the authorized connection time by evaluating the social closeness defined in Section 3.2.2 in an undirected friendship network from an SNS. Requests are sometimes blocked due to the limit of the request queue size or the number of connections to the owner's device.

Figure 3.5 shows the flow of the simulation. (1) When the owner and a guest user are located within a feasible communication range, the guest user sends a connection request to the owner's device. (2) The owner's device adds the request to a request queue. (3) The owner's device sends a request to the authentication server. (4) The authentication server determines the authorized connection time according to the social closeness between the guest user and owner of the tethering device.

Evaluation model

The parameters of the simulation are listed in Table 3.4. The detailed explanations of the parameters and components of the simulation are as follows.

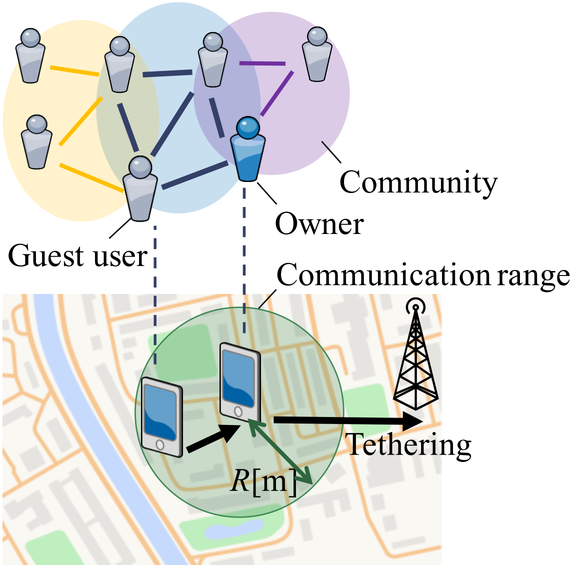


Figure 3.4: Evaluation scenario. (©2018 IEEE.)

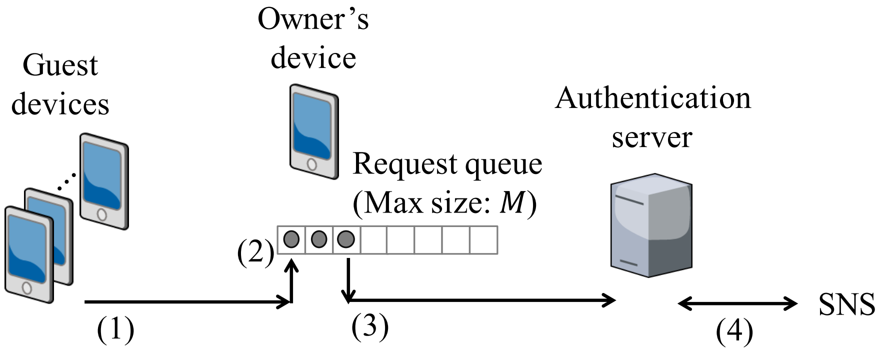


Figure 3.5: Simulation flow. (©2018 IEEE.)

Authentication server The authentication server receives connection requests from the users and determines the authorized connection time for each user. When the authentication server receives a request, it adds the request to the request queue. The size of the request queue is limited to M. If the authentication server receives a request when the request queue is full, the request will be blocked.

The authentication latency is defined as L. In this simulation, the actual measured value mentioned in Section 3.3.1 is used for L.

Shared device The tethering devices are shared with users and allow guest devices of authorized users to transmit a certain amount of data through it. The tethering devices can be accessed by up to N guest devices at the same time. Once the number of connected devices reaches N, all subsequent requests will be blocked until the authorized connection time of one of the connected devices expired.

Owner The relationships with the owner of the WiFi AP determine the authorized connection time for users, and 10% of the users in the dataset are randomly selected as candidates for owners. The simulation was conducted repeatedly for each owner selected from the candidates. The owners are assumed to stay in the i-th check-in location for $\min(T, t_{i+1} - t_i)$ minutes before she or he moves to the next check-in location, where t_i and t_{i+1} are the i-th and $(i + 1)$-th check-in times for the owner, respectively.

Guest users The guest users create connection requests and transmit data through the tethering device when authorized. The guest users are assumed to stay in the same location for a certain period as well as the owners.

Communication range A communication range is a range within a radius R from the current location of the owner. As the owner and guest users move around, when a guest user enters the communication range of the owner, the guest user makes a connection request to the owner's tethering device. On the other hand, when the owner or guest user leaves the current check-in location and the guest user is no longer within the communication range of the owner, all connection requests and connections to the owner are canceled at that point.

Authorized connection time The system determines the authorized connection time for each guest user according to the social closeness between the guest user and owner and the communities they belong to. If a guest user U_g is not blocked due to the limit of the request queue size or the number of connections at the owner's tethering device, the authorized connection time for U_g is defined as $\tau(U_g) = SC(U_o, U_g)\beta$, where U_o is the owner, $SC(U_i, U_j)$ is the social closeness between users U_i and U_j, as defined in (3.1), and β is a coefficient. In this simulation, the common neighbors, Jaccard Index, and Adamic-Adar Index defined in (2.1) in Section 2.1 are used as $E(U_o, U_g)$. The value for β is selected so that $\tau(U_g)$ does not exceed T for almost all user pairs.

Dataset

In this simulation, Brightkite datasets [73] were used as the data source of online social relationship. Brightkite is a popular online location-based social network. The friendship network of Brightkite was originally directed but was reconstructed as a network with undirected edges by only considering bi-directional edges [74]. To simplify the simulation, users who have at least one check-in in Japan are extracted. Friendships among the extracted users and communities detected by the Link communities algorithm [29] are used

Table 3.4: Simulation parameters.

Parameter	Value
Simulation period	Apr. 2008 – Oct. 2010
Radius of communication range (R)	100 m
Max. duration for users to stay at same location (T)	60 minutes
Max. no. of simultaneous connections (N)	5
Authentication latency (L), measured in Sect. 3.3.1	5.408 seconds
No. of guest users (V)	3,013
Max. size of request queue (M)	10
Mean of requested time in compared system (m)	60 minutes

to evaluate social relationships between device owners and guest users. The statistics about the extracted users are as follows.

Figure 3.6 shows the cumulative distribution function (CDF) curve of contact duration per contact. A contact starts when a guest user enters the communication range of the owner and ends when the guest user leaves it. The figure shows that about 50% of contacts were longer than 800 seconds. The maximum contact duration was limited to 3600 seconds because it cannot exceed T. Figure 3.7 shows the CDF curve of the number of devices connected to the tethering device over time. The maximum number of connected devices was limited to N. For about 70% of the time, the tethering device was connected by one guest user. Figure 3.8 shows the CDF curve of intervals of user check-ins. This figure illustrates that about 60% of check-ins were created within 6 hours from a previous check-in. Figure 3.9 is a double logarithmic chart that shows the number of users against the number of check-ins with a fitted curve having a slope of -0.79. When the number of check-ins was smaller than 100, the number of users decreased along the fitted curve as the number of check-ins increased, whereas when the number of check-ins was greater than 100, the number of users decreased faster than the fitted curve. Figure 3.10 is a double logarithmic chart that shows the number of user pairs against the number of common neighbors for all $(V-1)V/2$ user pairs with a fitted curve having a slope of -1.94. When the number of common neighbors was smaller than 60, the number of user pairs decreased along the fitted curve as the number of common neighbors increased, whereas when the number of common neighbors was greater than 60, the number of user pairs decreased faster than the fitted curve.

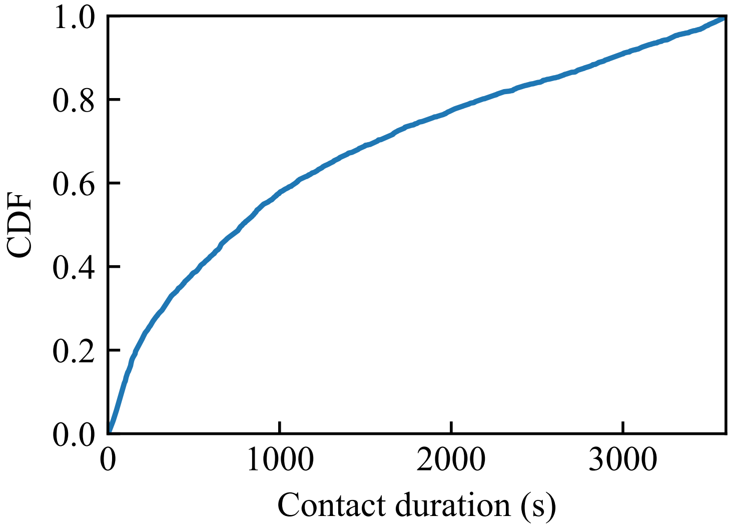


Figure 3.6: Contact duration per contact. (©2018 IEEE.)

Comparison system

The proposed system is compared with a system that does not evaluate online social relationships when it authenticates users. The authorized connection time is generated according to exponential distributions whose average is m. The guest user is allowed to access the tethering device for the same duration as the guest user requested until she or he leaves the communication range of the owner, regardless of the social closeness between the owner and guest user. The proposed system was compared with the comparison system based on the average actual connected duration per connection request.

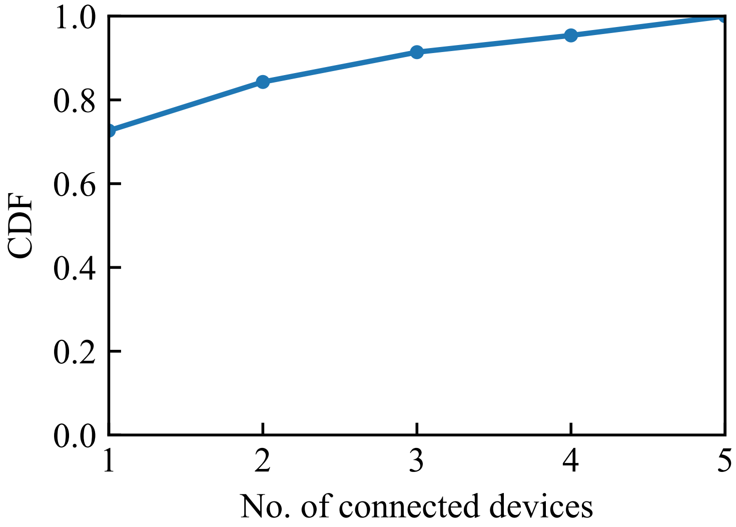


Figure 3.7: No. of connected devices. (©2018 IEEE.)

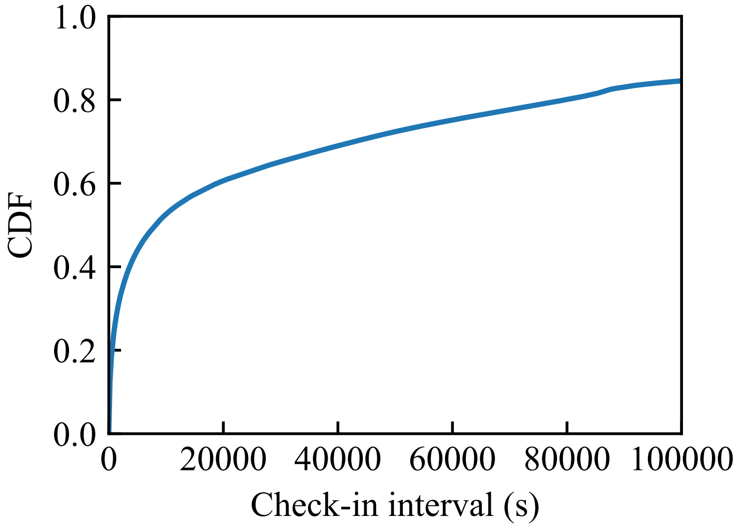


Figure 3.8: Check-in interval. (©2018 IEEE.)

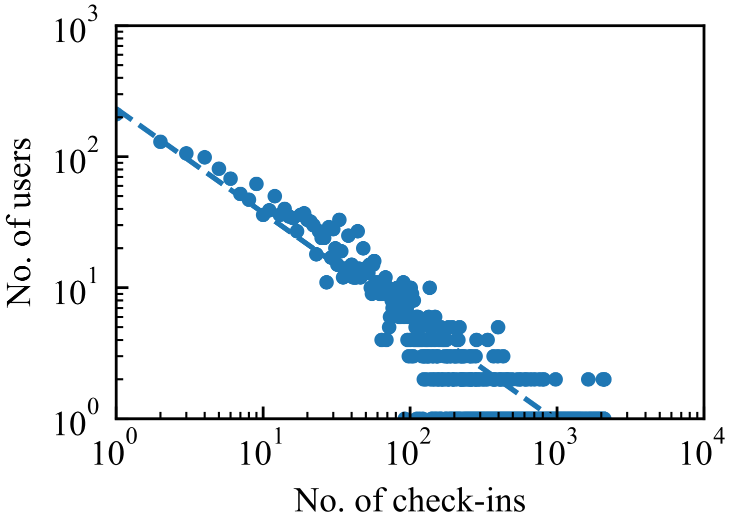


Figure 3.9: No. of users vs. no. of check-ins. (©2018 IEEE.)

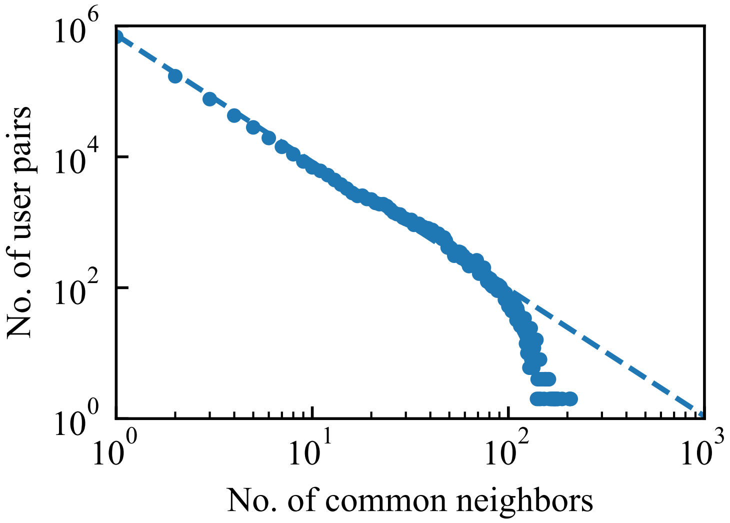


Figure 3.10: No. of user pairs vs. no. of common neighbors. (©2018 IEEE.)

Results

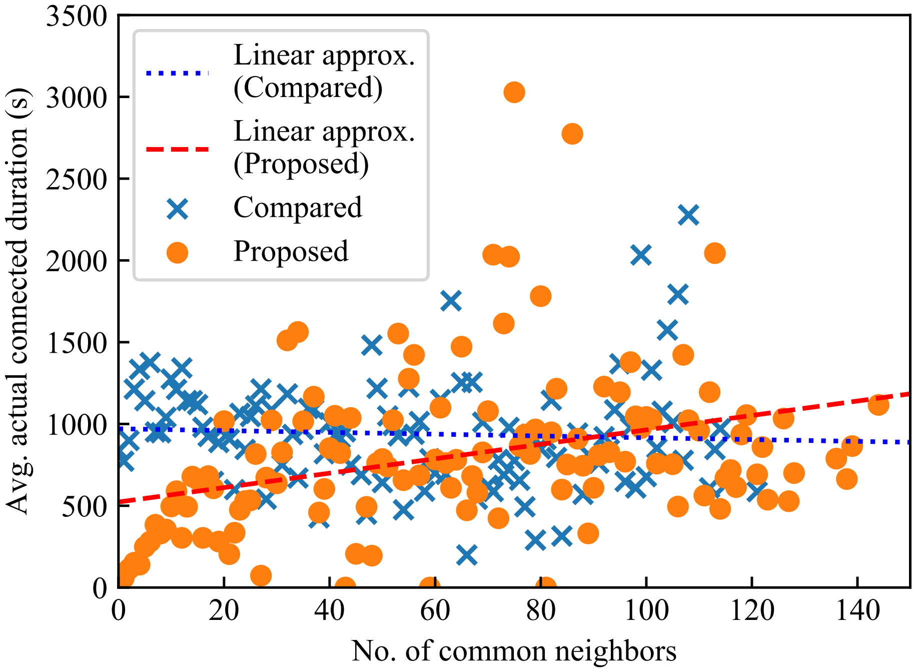


Figure 3.11: Avg. actual connected duration vs. social closeness (common neighbors, $\beta = 1$). (©2018 IEEE.)

The following two points can be observed from the results; i) the proposed system limits the resource usage for guest users who are not as close to the device owners, and ii) the overhead of the authentication process in the system does not interfere with the resource sharing with guest users who are close to the device owners.

Figures 3.11, 3.12 and 3.13 plot the average actual connected duration per connection requests against the number of common neighbors, Jaccard Index, and Adamic-Adar Index, respectively. According to the linear approximate line, as the number of common neighbors increased, the average actual connected duration on the proposed system also increased, while there was no significant change on the comparison system.

In Fig. 3.11, when the number of common neighbors was smaller than 90,

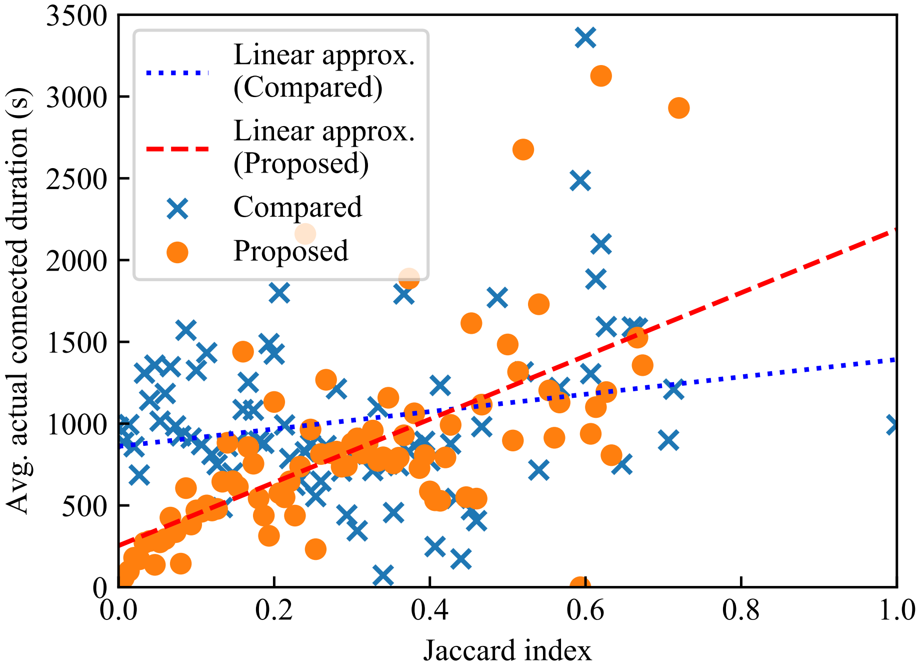


Figure 3.12: Avg. actual connected duration vs. social closeness (Jaccard Index, $\beta = 150$). (©2018 IEEE.)

the guest users had a shorter actual connected duration on the proposed system than the comparison system. As seen in Fig. 3.6, about 50% of contacts were longer than 800 seconds. However, according to the linear approximate line, the average actual connected duration of the proposed method was shorter than 800 seconds. This is because the average actual connected duration was properly limited by $\tau(U_g)$. This indicates that the proposed system properly limited the authorized level of resource usage for unfamiliar guest users. On the other hand, when the number of common neighbors was greater than 90, the guest users had longer actual connected duration on the proposed system than the comparison system. This is because the authentication latency L, which was set to the actual measured value mentioned in Section 3.3.1, was much shorter than the average connected duration. Therefore, the proposed system allowed socially close guest users to use the shared devices with only a little interference by its authentication overhead. As a result, points i) and ii) mentioned earlier in this section can be observed from Fig. 3.11.

Figures 3.12 and 3.13 show the same trend as in Fig. 3.11. According to the linear approximate line, when Jaccard Index or Adamic-Adar Index was small, the guest users had a shorter actual connected duration on the proposed system than the comparison system. On the other hand, when Jaccard Index or Adamic-Adar Index was large, the average actual connected duration of the proposed system was longer than that of the comparison system. Therefore, points i) and ii) can also be observed from Figs. 3.12 and 3.13.

3.4 Chapter summary

This chapter proposed a system that uses online social relationships to meet device owners' demand for resource management for altruistic device sharing. The proposed system enables device owners to reduce their costs of device sharing with users according to the social closeness between the device owners and guest users. A prototype system was implemented to confirm that the proposed system can be fully implemented as an actual working system and measure the authentication overhead of the proposed system. A simulation was conduced using this overhead measured on the prototype and a large-scale dataset of a real social network. The simulation verified that i) the proposed

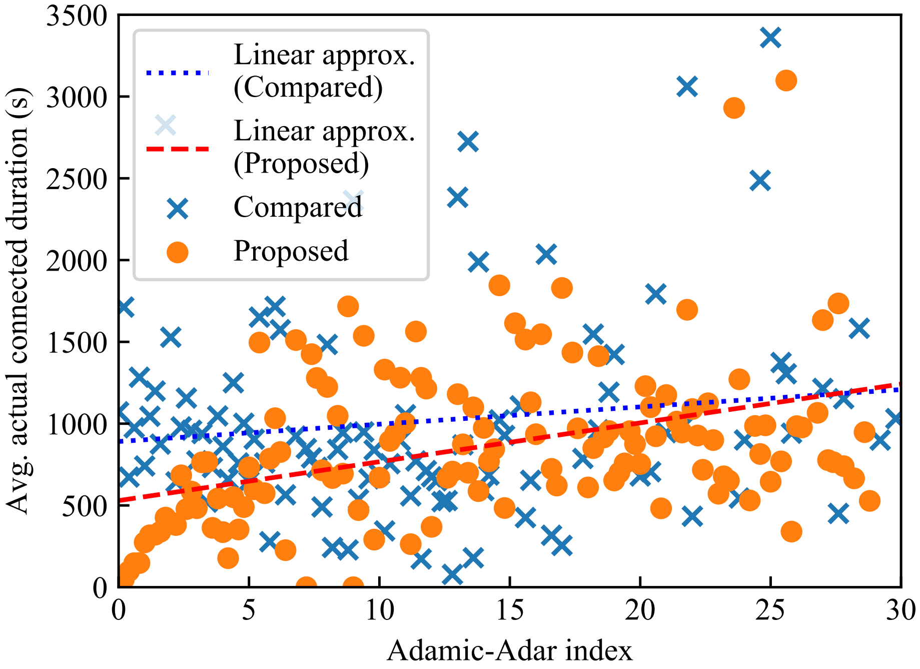


Figure 3.13: Avg. actual connected duration vs. social closeness (Adamic-Adar Index, $\beta = 5$). (©2018 IEEE.)

system limits the resource usage for guest users who are not as close to the device owners, and ii) the overhead of the authentication process in the system does not interfere with the resource sharing with guest users who are close to the device owners.

Chapter 4

Prioritization of mobile IoT data transmission based on data importance extracted from machine learning model

4.1 Overview

The increasing impact of social problems related to road traffic is a major concern facing our future society. Traffic accidents are still a major problem in many societies today. According to a report on road safety by the world health organization (WHO), road traffic injuries are currently estimated to be the ninth leading cause of death across all age groups globally and are predicted to become the seventh leading cause of death by 2030 [75]. It also states that 3% of the global gross domestic product (GDP) is estimated to be lost as a result of road traffic deaths and injuries. Road traffic congestion is another serious problem in many countries. A report by the Centre for Economics and Business Research suggests that the total economy-wide cost across four advanced countries (the UK, France, Germany, and the USA) was $200.7 billion in 2013, and is forecasted to rise to $293.1 billion by 2030 [76].

Predicting real-time spatial information from data collected by mobile IoT sensors is one solution to solve the social problems related to road traffic [77].

Mobile IoT devices such as smart cars (including autonomous cars), smartphones, wearable devices, and unmanned aerial vehicles (UAVs) play a major role in such an application: namely, they work to collect data. Some studies have discussed algorithm design for collecting data from sensors on vehicles using mobile crowdsensing [78] [79]. The data collected by mobile IoT devices are uploaded to edge servers, which process the uploaded data and apply machine learning techniques to predict real-time spatial information such as road-traffic volume, optimal travel path, and precise positions of pedestrians and cyclists. Real-time spatial information prediction is in demand for many services. An example service is the autonomous driving support system, which gathers real-time data from onboard sensors and provides exact location and relation to other road users [80]. The cyber-physical system (CPS), in which the real-time spatial information prediction system is included, is increasingly in demand. The market was worth $18 billion in 2017 and is likely to grow by 8.7% annually for the next ten years [81].

However, mobile IoT devices for real-time spatial information prediction collect an enormous amount of upstream data — much more than can be collected through the uplink bandwidth in mobile networks. Mobile IoT devices collect images, videos, or light detection and ranging (LiDAR) [82] data continuously, and it is impossible to collect all of such data through the uplink bandwidth of LTE or LTE-A networks today. Even with 5G networks, it is impossible to collect all of the high-resolution images, videos, and LiDAR data.

Cluster-based data aggregation reduces data transmission by clustering wireless sensors and aggregating raw data from each cluster before transmitting them to destined targets [83] [84] [85] [86]. Sensors clustered into one cluster are usually located nearby each other, so collected data from these sensors are correlated and thus redundant to some extent. Cluster-based data aggregation eliminates this redundancy, thereby reducing the volume of data transmission. This approach focuses mainly on redundancy in data; no previous work has successfully reduced the volume of transmitted data used as input for real-time prediction while maintaining the prediction accuracy of real-time spatial information.

This chapter proposes an IoT device control system that reduces the volume of transmitted data used as input for real-time prediction while maintaining the prediction accuracy of real-time spatial information. The main contribution of this chapter is that the proposed system prioritizes the transmissions of data collected by mobile IoT devices on the basis of the “importance of data” extracted from the machine learning model for prediction. The importance of data is a metric of how much the data collected by mobile sensors will contribute to the prediction accuracy of real-time spatial information. Feature selection has been widely used to extract the importance of data from the machine learning model. Feature selection methods were originally used to reduce computation time, improve prediction performance, or provide a better understanding of the data in machine learning or pattern recognition applications. Feature selection methods were also used to reduce communication overhead in distributed learning [87]. Unlike those conventional usages, the proposed system uses feature selection methods to control the data transmission of mobile IoT devices with priority. In this work, two performance evaluations are performed using real-world datasets, with each one assuming a different scenario. These evaluations use a random forest regressor [44] as the machine learning model for prediction and the impurity method [43] and perturb method [45] as feature selection methods. The results of these evaluations show that the proposed system reduces the volume of input data transmission for real-time prediction compared with benchmark methods while achieving the same prediction accuracy.

The rest of this chapter is organized as follows. Section 4.2 presents the problem formulation of this study and the details of the proposed system. Sections 4.3 and 4.4 provide performance evaluations with scenarios of road-traffic volume prediction and mobility demand prediction, respectively. This chapter is concluded in Section 4.5 with a brief summary.

4.2 Proposed system design

The proposed system prioritizes the transmissions of data collected by mobile IoT devices on the basis of the importance of data extracted from the machine learning model for prediction using feature selection. The importance of data extracted from the machine learning model using feature selection is a metric of how much the collected data by mobile sensors will contribute to the

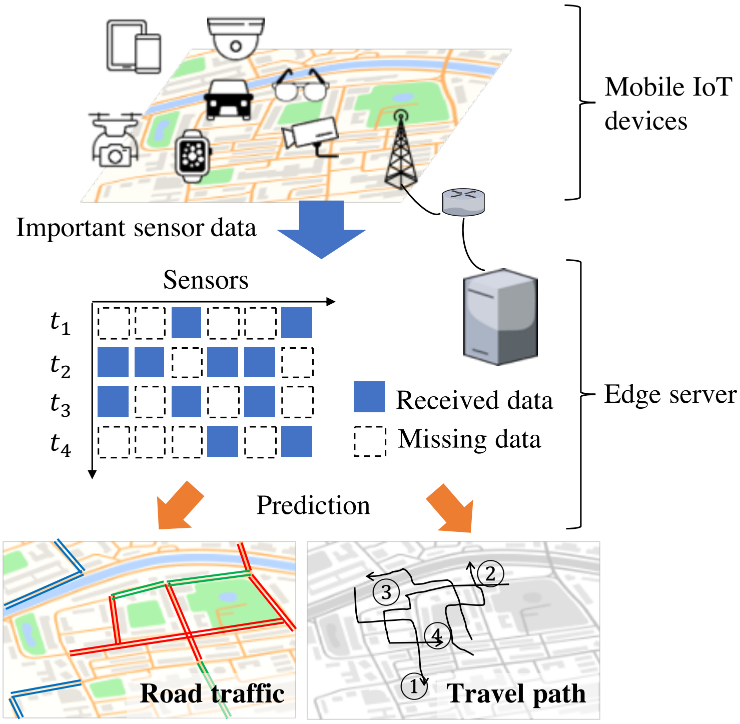


Figure 4.1: System overview. (©2019 IEEE.)

prediction accuracy of real-time spatial information. Reducing transmission of less important data according to the importance of data obtained from feature selection methods enables the proposed system to reduce the volume of transmitted data used as input for real-time prediction while maintaining the prediction accuracy.

4.2.1 Application scenario

The overview of the proposed system is shown in Fig. 4.1. This work assumes a system that provides users with real-time spatial information based on data collected from mobile IoT devices. The proposed system prioritizes data on mobile IoT devices on the basis of data importance extracted from the machine learning model for prediction, which enables it to reduce the total data traffic for real-time prediction while maintaining prediction accuracy.

The proposed system consists of two main components: mobile IoT devices and an edge server. Mobile IoT devices (such as probe vehicles, smartphones,

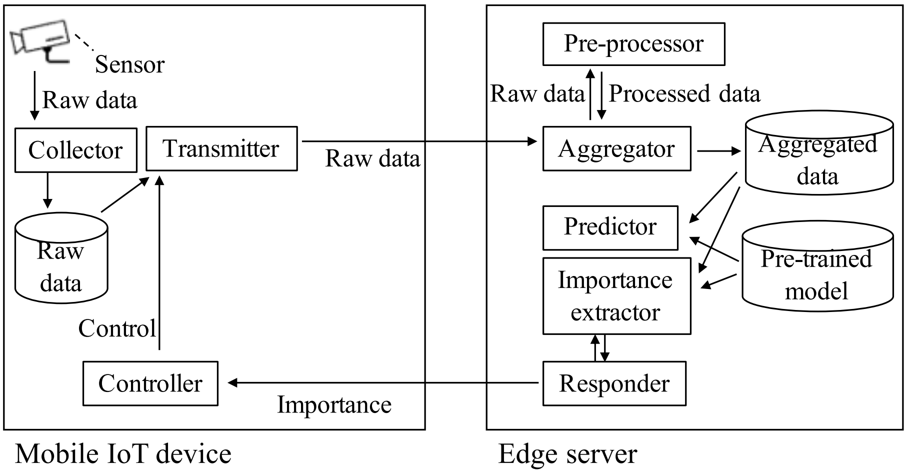


Figure 4.2: System design. (©2019 IEEE.)

and UAVs) prioritize collected data and send high importance input data for prediction to the edge server. The edge server aggregates the data received from mobile IoT devices, complements the missing parts of the data, and performs prediction.

Note that the machine learning model has been trained in advance with all the available data collected by mobile IoT devices. This assumption is acceptable because this work aims to reduce the volume of data used as input for real-time prediction. Furthermore, since the time requirement of training data is not strict, they can be collected as a background process through mobile networks during the off-peak period or through other communication networks with sufficient bandwidth.

4.2.2 System model

A detailed view of the proposed system is shown in Fig. 4.2.

Mobile IoT device Mobile IoT devices (such as probe vehicles, smartphones, and UAVs) continuously collect data at a specific sampling interval. Part of the collected data is sent to an edge server for prediction. To decide whether a data should be sent or not, the controller fetches the importance of the block where the mobile IoT device is currently located. The controller

Table 4.1: Notations.

	Description
$D(\bar{I})$	Set of data received from mobile IoT devices under data importance threshold $\bar{I}$
$S(\cdot)$	Size of data
$A(\cdot)$	Prediction accuracy
$\bar{A}$	Required prediction accuracy
X	Input variable of the machine learning model for prediction ($T \times N_B$ matrix)
y	Output variable of the machine learning model for prediction (vector of N_B elements)
T	No. of time slots used in one prediction
N_B	No. of blocks
$F_{t,b}$	Feature importance of (t, b) element of the input variable
I_b	Importance of the block b
$\bar{I}$	Threshold for importance of transmitted data

orders the transmitter to send the data or not based on the importance of the current block received from the edge server.

Edge server The edge server receives collected data from mobile IoT devices, aggregates and preprocess the data, and predicts the real-time spatial information of the next time slot. The aggregator and pre-processor on the edge server receive data from the transmitters of several mobile IoT devices, pre-process the data, and complete the missing parts of the data. The predictor predicts real-time spatial information. The importance extractor extracts the importance of blocks from the pre-trained machine learning model for prediction. The responder sends the importance of blocks to each mobile IoT device.

4.2.3 Control methods

The control procedure of the proposed system consists of five processes: 1) pre-training of the machine learning model, 2) calculation of the importance of blocks, 3) control of data transmission, 4) aggregation of transmitted data, and 5) prediction. 1) and 2) are preprocessing, which is performed once before the first time slot begins. 3), 4), and 5) are performed in each time slot. Table 4.1 lists the notations in this chapter.

Pre-training of machine learning model

The machine learning model is trained on the edge server in advance before the first time slot begins.

The proposed system considers the prediction of future real-time spatial information as a regression task. Regression, in general, is a type of task that estimates a numerical value given some input [88]. To solve the task, the learning algorithm is asked to learn a function that maps an input variable to an output variable. The proposed system uses a supervised machine learning model to solve the task. Supervised learning algorithms, in general, deal with a training dataset that contains a set of data and a label or target associated with each of the data [88].

In the proposed system, the machine learning model for prediction receives the aggregated past sensor data collected from mobile IoT devices in each block as an input and calculates the future real-time spatial information as output. The machine learning model for prediction receives an input variable X, which consists of aggregated sensing data collected in the last few time slots, and predict output variable y, which is the real-time spatial information of each block in the next time slot. The input variable X of the machine learning model for prediction is a $T \times N_B$ matrix, where T is the number of time slots used in one prediction and N_B is the number of blocks. Each row of the input matrix represents the aggregated sensor data collected in N_B blocks at $T, T-1, \ldots, 1$ slots ago, respectively. The output y of the machine learning model for prediction is a vector of N_B elements. Each element of the output vector represents the real-time spatial information of each block in the next time slot.

As mentioned in Section 4.1, to train a machine learning model for prediction, an adequate amount of past sensing data should be collected from mobile IoT devices as training data, but it does not necessarily need to be collected in real time. Thus, the training data can be collected when the network is off-peak, such as at night or when cars or drones are stopped in parking lots or depots.

Calculation of importance of blocks

The importance of a block is calculated from the pre-trained machine learning model for prediction on the edge server using a feature selection method. The importance of block I_b is defined as $I_b = \sum_{t=1}^{T} F_{t,b}$, where $F_{t,b}$ is the feature importance of the (t, b) element of the input matrix. $F_{t,b}$ is calculated from the pre-trained model using the feature selection method.

Control of data transmission

In each time slot, mobile IoT devices decide whether to transmit the data observed in the time slot based on the importance of the data. The importance of data corresponds to the importance of the block where the data was observed. Mobile IoT devices transmit the data if $I_b \geq \bar{I}$, where I_b is the importance of the block that includes the current location of the device and $\bar{I}$ is a constant that defines the minimum importance for the data to be transmitted. I_b and $\bar{I}$ are obtained from the edge server.

The proposed system can control the volume of the transmitted data and prediction accuracy through $\bar{I}$. The volume of the transmitted data in a single time slot can be described as

$$\sum_{d \in D(\bar{I})} S(d), \tag{4.1}$$

where $D(\bar{I})$ is the set of transmitted data from mobile IoT devices and $S(d)$ is the size of data d. $D(\bar{I})$ includes data from a mobile IoT device if and only if $I_b \geq \bar{I}$, where I_b is the importance of the block in which that device is located. The prediction accuracy in a time slot can be described as

$$A(D(\bar{I})), \tag{4.2}$$

where $A(\cdot)$ is the prediction accuracy when given a set of sensing data from mobile IoT devices. The prediction accuracy depends on the data received from mobile IoT devices in the time slot.

Aggregation of transmitted data

The data collected from mobile IoT devices are aggregated to form an input matrix X for the machine learning model for prediction. The aggregation is needed because the proposed system does not always collect exactly one data from each block. The number of data collected from each block varies depending on the importance of the block and the number of mobile IoT devices in the block. An example of the aggregation process can be found in Section 4.3.2.

Prediction

The proposed system uses the pre-trained model to predict future spatial-information. In each time slot, the model takes the aggregated sensing data as an input X and predicts the real-time spatial information of the next time slot as an output y.

4.3 Performance evaluation by road-traffic volume prediction

4.3.1 Evaluation scenario

An evaluation was performed to verify the effectiveness of the proposed system described in Section 4.2, which reduces the total traffic for real-time prediction transmitted from mobile IoT devices while maintaining the prediction accuracy. This evaluation examines the relationship between the amount of transmitted data and prediction accuracy described in Eqs. (4.1) and (4.2) respectively for several $\bar{I}$.

This performance evaluation focuses on a specific application that provides human or robotic drivers with road-traffic information predicted from sensing data collected by onboard cameras or LiDARs on probe vehicles. Parameters used in this evaluation are listed in Table 4.2.

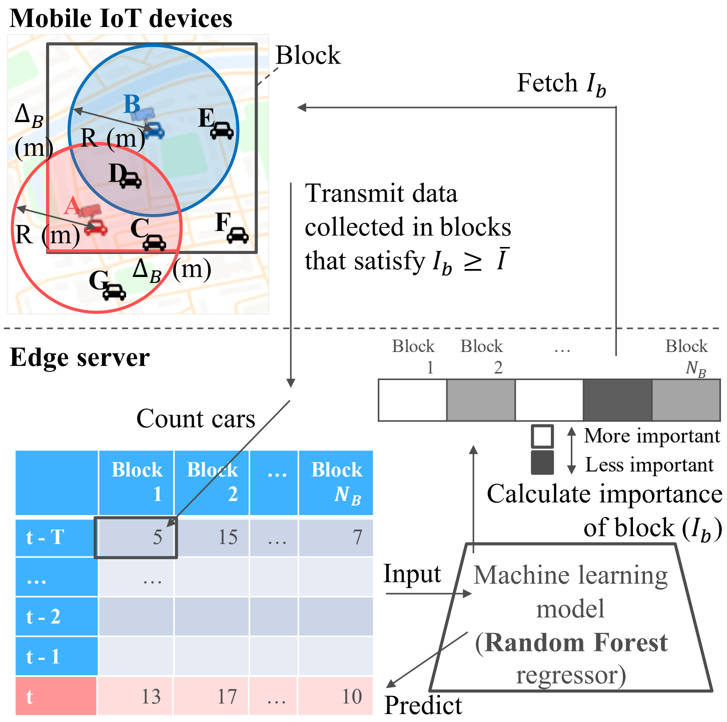


Figure 4.3: Evaluation model for performance evaluation by road-traffic volume prediction. (©2019 IEEE.)

Table 4.2: Parameters for performance evaluation by road-traffic volume prediction.

Parameter	Value
Sampling interval (Δ_T)	1 (minute)
Sampling intervals in one time slot (T)	60
Radius in which probe vehicles can detect cars (R)	50 (m)
Size of block (Δ_B)	1000×1000, 500×500 (m^2)
No. of blocks (N_B)	144, 576
Percentage of probe vehicles (P)	20 (%)
No. of estimators in random forest (N_E)	100

4.3.2 Evaluation model

Figure 4.3 shows the evaluation model used for this evaluation. It consists of mobile IoT devices and an edge server.

Pre-training of machine learning model

The random forest regressor model in the scikit-learn library [89] is used as the machine learning model for prediction for this evaluation. The input X of the model is $T \times N_B$ matrix, where T is the number of data samples in one time slot and N_B is the number of blocks. Each element x_{ij} in the matrix represents the aggregated road-traffic in block j at time slot $t - i$, where t is the current sampling time. The output y of the model is the road-traffic of each block at sampling time t.

The model is trained with road-traffic data of all 536 taxies before the evaluation. The road-traffic data of all 25 days is split into the first 20 days and the last five days for training and evaluation, respectively. The details on the dataset are described in Section 4.3.3.

Calculation of importance of blocks

Two feature selection methods are used to calculate the importance of blocks: the impurity method and the perturb method. The impurity method calculates feature importance on the basis of the ‘impurity’ index used in decision tree models [43]. The impurity method in this evaluation is implemented by the `feature_importances_` function of the random forest regressor of scikit-learn. By applying this function, the importance of each input feature, i.e., the importance of each element x_{ij}, is obtained. To simplify the evaluation, the importance of blocks are calculated by taking the sum for i. The perturb method calculates feature importance by adding noise to the subset of input features and examining the increase of error [45]. The perturb method calculates the importance of block j using root mean square error (RMSE) by

$$(RMSE(\hat{y}', y) - RMSE(\hat{y}, y))^2, \tag{4.3}$$

where y is the number of cars in blocks, $\hat{y}$ is the predicted value of y, and $\hat{y}'$ is the predicted value when the input values of block j in the training data are multiplied by 1.5.

Control of data transmission

In each time slot, probe vehicles transmit the collected sensing data if and only if $I_b \geq \bar{I}$, where I_b is the importance of the block in which a probe vehicle is currently located. Probe vehicles know the I_b of each block in advance.

The number of cars can be detected from sensing data collected by onboard cameras or LiDARs using an object detection algorithm at the pre-processor on the edge server. In this evaluation, this process is streamlined and the number of cars is obtained directly from an existing dataset.

Aggregation of transmitted data

It is assumed that an aggregator on the edge server receives raw sensing data from probe vehicles and a pre-processor on the edge server identifies cars that are running around each probe vehicle. The number of detected cars in the block at the sampling time is defined as the size of the set plus 1. If multiple probe vehicles are in the block at the sampling time, this number is defined as the size of the union of sets of cars plus the number of probe vehicles. If no probe vehicles are in the block at the sampling time, zero-filling is used to complete the missing parts of data.

Prediction

Prediction is performed in each time slot using the pre-trained model described in Section 4.3.2.

4.3.3 Dataset

A trace set of the mobility data of taxi cabs in San Francisco [90] is used in this evaluation. The dataset includes the location logs of 536 taxies for 25 days. N_B blocks in total are positioned in a rectangle area, as shown in Fig. 4.4. Since the logs are not necessarily recorded every Δ_T minutes, taxies are assumed to travel straight with constant velocity, and locations at every Δ_T minute are interpolated. Probe vehicles are selected randomly from a total of

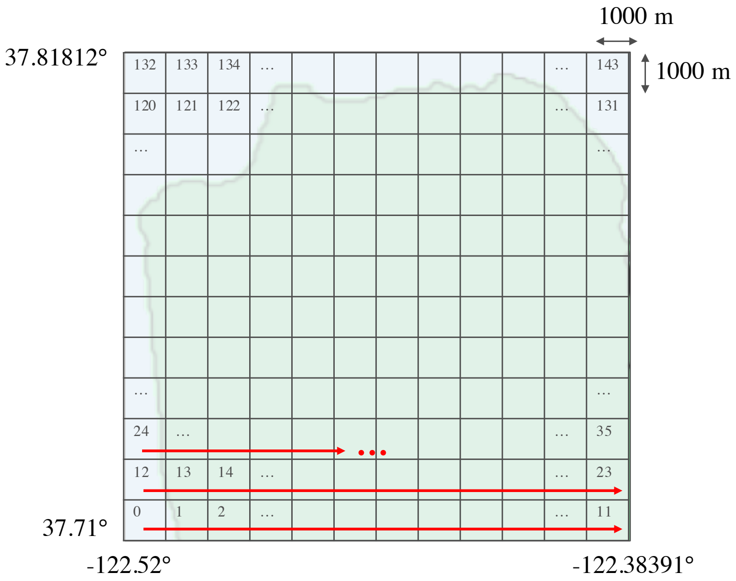


Figure 4.4: Layout of blocks ($\Delta_B = 1000 \times 1000$ (m^2)). (©2019 IEEE.)

536 taxies at a ratio of P. The number of cars detected by probe vehicles in each block at each sampling time is calculated as described in the previous section. A block that contains the taxi company depot is ignored because the block does not seem to generate data appropriate for evaluation. Figure 4.5 shows the average number of detected cars in each block.

4.3.4 Metrics and benchmarks

This evaluation verifies that the proposed system reduces the amount of transmitted data used as input for real-time prediction compared with three benchmark methods when they achieve the same prediction accuracy by examining the relationship between the amount of transmitted data and prediction accuracy described in Eqs. (4.1) and (4.2) respectively. The prediction error and the total amount of data transmission are calculated for each $\bar{I}$.

To evaluate the prediction error, the root mean squared log error (RMSLE) [91] function is used. RMSLE is given by

$$RMSLE = \sqrt{\frac{1}{N_B}\sum_{b=1}^{N_B}(\log(y_b + 1) - \log(\hat{y_b} + 1))^2}, \tag{4.4}$$

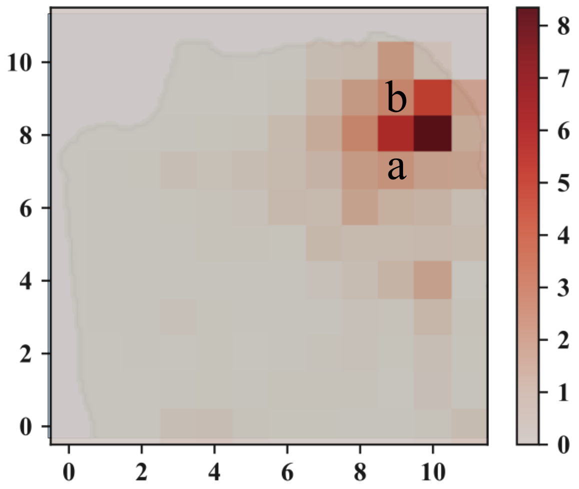


Figure 4.5: Average no. of detected cars in each block. (©2019 IEEE.)

where y_b is the number of cars running in block b and $\hat{y_b}$ is the predicted value of y_b.

To evaluate the total amount of data transmission, the normalized total amount of transmitted data is used. The normalized total amount of transmitted data r is the ratio of the total amount of sensing data transmitted to the total amount of sensing data collected by probe vehicles.

This evaluation uses three benchmark methods: random, uniform, and volume-based. The random method selects which block to use at random. This is a reasonable method because, in general, the data transmitted by mobile IoT sensors are usually dropped randomly when network capacity is limited. The uniform method selects n_B blocks out of the total N_B blocks uniformly. To select blocks uniformly, the uniform method spirally assigns numbers $0 \leq k < N_B$ to each block. The set of numbers of selected blocks is decided by $\{k = \lfloor (nN_B)/n_B \rfloor \mid 0 \leq \exists n < n_B\}$. The volume-based method selects blocks with the top n_B largest average road-traffic volume in the training dataset.

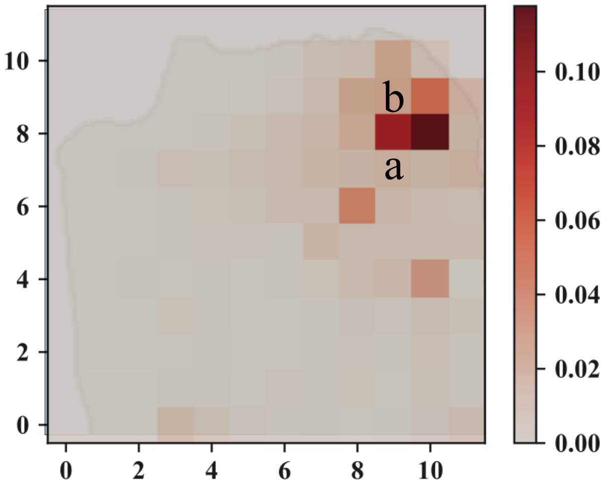


Figure 4.6: Extracted importance of each block for road-traffic volume prediction (impurity). (©2019 IEEE.)

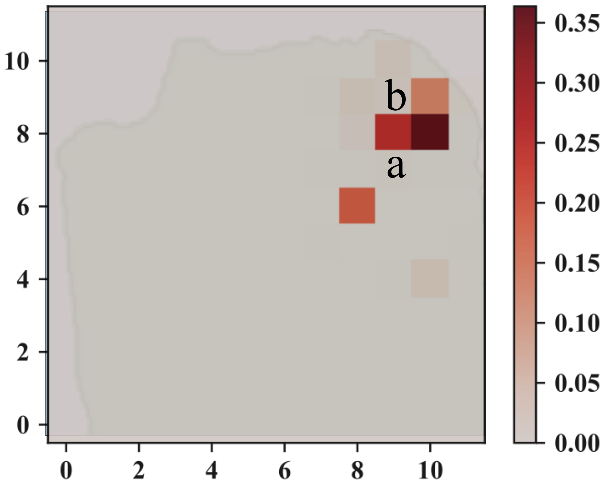


Figure 4.7: Extracted importance of each block for road-traffic volume prediction (perturb). (©2019 IEEE.)

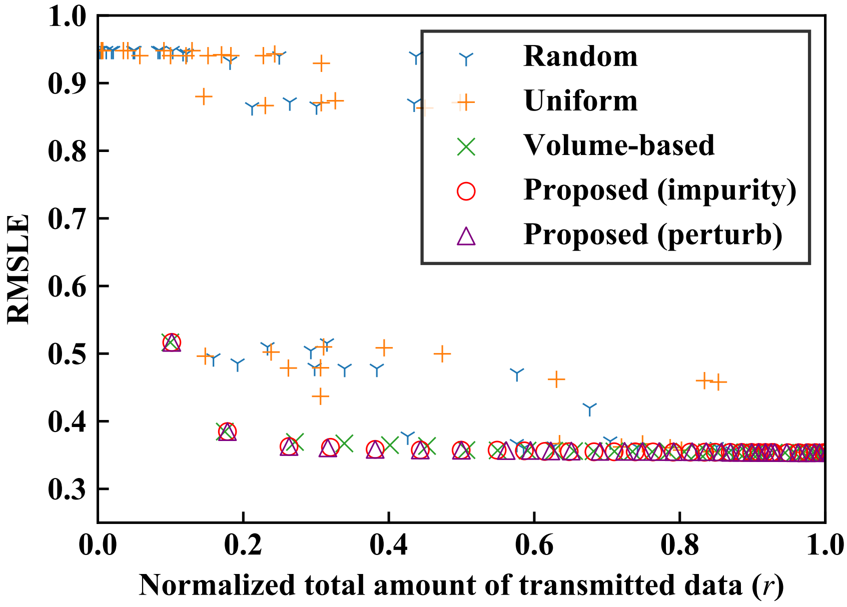


Figure 4.8: Prediction error vs. normalized total amount of transmitted data for road-traffic volume prediction ($\Delta_B = 1000 \times 1000$ (m^2)). (©2019 IEEE.)

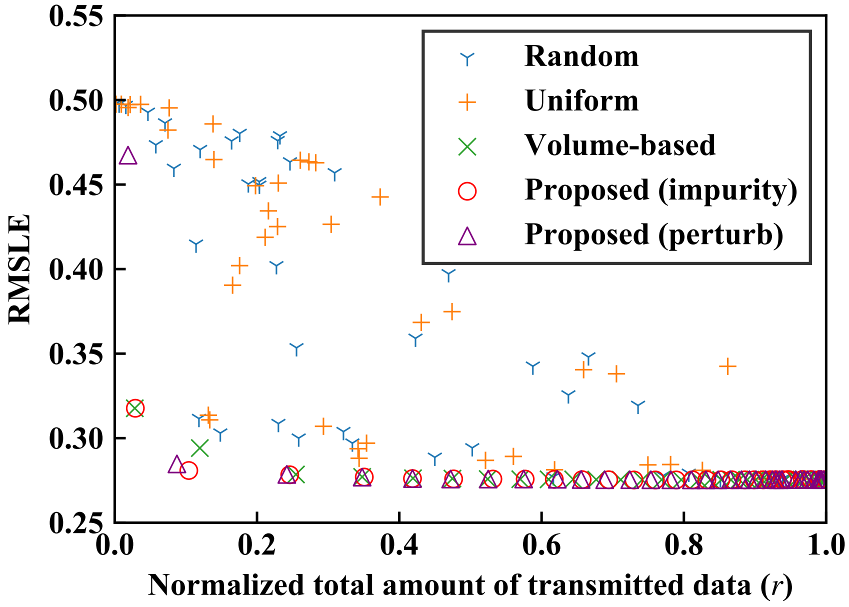


Figure 4.9: Prediction error vs. normalized total amount of transmitted data for road-traffic volume prediction ($\Delta_B = 500 \times 500$ (m^2)). (©2019 IEEE.)

4.3.5 Results

Figures 4.6 and 4.7 show the importance of each block extracted from the random forest regressor by using the impurity and perturb feature selection methods, respectively. Compared with the average number of detected cars in each block in Fig. 4.5, blocks with a large road-traffic volume tend to be important. However, the blocks whose adjacent blocks have large road-traffic volume (e.g., a, b) tend to be less important compared to their own road-traffic volume. This is because the data from two adjacent blocks are redundant to some extent. In general, the road-traffic volumes of two adjacent blocks correlate with each other. The impurity and perturb methods reflect this principle and avoid assigning high importance to two adjacent blocks in order to eliminate that redundancy.

Figure 4.8 shows the prediction error against the normalized total amount of transmitted data when the block size is $\Delta_B = 1000 \times 1000$. RMSLE is larger as r is smaller for all the methods. This describes the trade-off between the amount of data available for prediction and the accuracy of prediction. RMSLE of the random and uniform methods fluctuates as r changes. This is because only a small number of blocks mainly contribute to the prediction accuracy, and thus RMSLE of the random and uniform methods depends greatly on whether those blocks are selected or not. In contrast, RMSLE of the impurity and perturb methods is stably small for a wide range of r. This is because the proposed system with the impurity or perturb methods always prioritizes the data from probe vehicles in important blocks, which contributes to the prediction accuracy. RMSLE of the impurity and perturb methods is better than that of the volume-based method for a wide range of r. This is because, as observed in Figs. 4.5 – 4.7, blocks with high average road-traffic volume do not always have high importance. Prioritizing the transmissions on the basis of the average road-traffic volume of each block leads to redundant data transmission. By using the importance of blocks, the impurity and perturb methods avoid redundant data transmission, and thus the proposed system can achieve better RMSLE than that of the volume-based method.

Figure 4.9 shows the prediction error against the normalized total amount of transmitted data when the block size is $\Delta_B = 500 \times 500$. The same obser-

Table 4.3: Parameters for performance evaluation by mobility demand prediction.

Parameter	Value
Sampling interval (Δ_T)	1 (minute)
Sampling intervals in one time slot (T)	60
Size of block (Δ_B)	1000×1000, 500×500 (m^2)
No. of blocks (N_B)	144, 576
No. of estimators in random forest (N_E)	100

vations can basically be obtained as in Fig. 4.8, while the scale of RMSLE of Fig. 4.9 is smaller as a whole than that of Fig. 4.8. This is because, in general, the prediction error tends to be small when the scale of predicted values is small. The scale of predicted values in this evaluation was smaller when $\Delta_B = 500 \times 500$ than when $\Delta_B = 1000 \times 1000$ because smaller blocks have smaller road-traffic volume.

4.4 Performance evaluation by mobility demand prediction

4.4.1 Evaluation scenario

This evaluation focuses on a specific application in which the system predicts the number of pickups by taxis on the basis of people detection from sensing data collected by probe vehicles. The purpose of this evaluation is the same as that of performance evaluation in Section 4.3. Parameters used are listed in Table 4.3.

4.4.2 Evaluation model

Pre-training of machine learning model

The random forest regressor model in the scikit-learn library [89] is used as the machine learning model for prediction for this evaluation. The shape of input X of the model is the same as the input described in Section 4.3, except that each element x_{ij} in the matrix represents the number of pickups in block j at time slot $t - i$, where t is the current sampling time. The output y of the model is the number of pickups of each block at sampling time t.

The random forest regressor is trained with the pickup log data of 536 taxies before the evaluation. The data is split into the first 20 days for training and last five days for evaluation. The details of the dataset are described in Section 4.4.3.

Calculation of importance of blocks

The impurity and perturb methods described in Section 4.3.2 are also used as feature selection methods for the proposed system in this evaluation.

Control of data transmission

The number of pickups by taxis can be detected from sensing data collected by onboard cameras or LiDARs using an object detection algorithm at the pre-processor on the edge server. In this evaluation, this process is streamlined and the number of pickups is obtained directly from an existing dataset as in Section 4.3.

The same as in Section 4.3.2, probe vehicles decide whether or not to transmit the collected data in each time slot.

Aggregation of transmitted data

It is assumed that an aggregator on the edge server receives raw sensing data and pickup logs from taxies and a pre-processor on the edge server extracts useful information to predict mobility demand. From the extracted information and pickup logs, the aggregator counts the number of pickups in each block at each time slot. If no probe vehicles are in the block at the sampling time, zero-filling is used to complete the missing parts of data.

Prediction

Prediction is performed in each time slot using the pre-trained model described in Section 4.4.2.

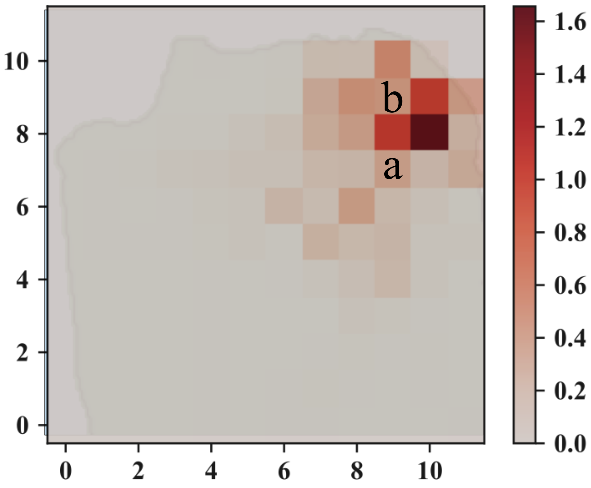


Figure 4.10: Average no. of pickups in each block per minute. (©2019 IEEE.)

4.4.3 Dataset

A trace set of the mobility data of taxi cabs in San Francisco is used in this evaluation, the same as in 4.3. The mobility log includes occupancy data that represents whether or not a taxi has passengers. This evaluation uses the occupancy of taxies as well as the mobility traces of taxies included in the dataset. This evaluation considers that a pickup occurred when the occupancy value of the log changed from 0 (not-occupied) to 1 (occupied).

The evaluation area is split into N_B blocks, as described in Section 4.3, and the location of taxies is interpolated every Δ_T minute. Figure 4.10 shows the average number of pickups in each block.

4.4.4 Metrics and benchmarks

This evaluation uses the same metrics and benchmark method as Section 4.3.

4.4.5 Results

Figures 4.11 and 4.12 show the importance of each block extracted from the random forest regressor by using the impurity method and the perturb method,

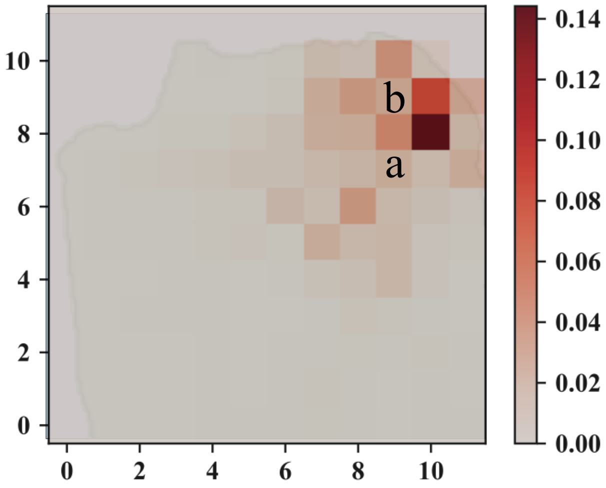


Figure 4.11: Extracted importance of each block for mobility demand prediction (impurity). (©2019 IEEE.)

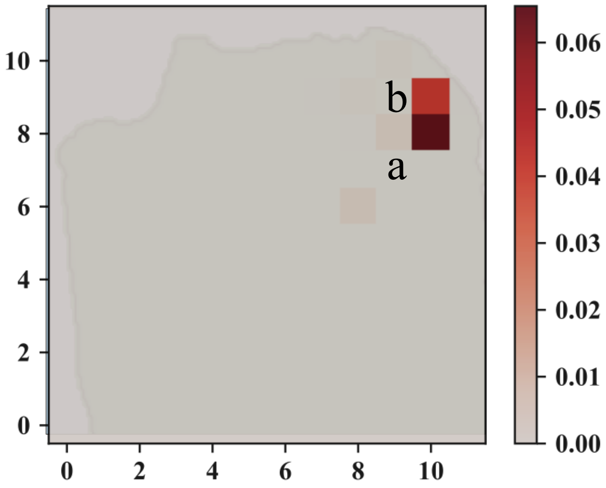


Figure 4.12: Extracted importance of each block for mobility demand prediction (perturb). (©2019 IEEE.)

respectively. As observed in Section 4.3.5, there is a difference between the number of pickups in Fig. 4.10 and the importance of blocks in Figs. 4.11 and 4.12.

Figures 4.13 and 4.14 show the prediction error against the transmission rate when the block size is $\Delta_B = 1000 \times 1000$ and $\Delta_B = 500 \times 500$, respectively. RMSLE is larger when the transmission rate is smaller for all the methods, which is the same as the trend observed in Figs. 4.8 and 4.9. In Figs. 4.13 and 4.14, when $r < 0.2$, RMSLE of the proposed system has a relatively larger value than RMSLE in other ranges of the transmission rate r. In contrast, in Figs. 4.8 and 4.9, when $r < 0.1$, RMSLE of the proposed system has a relatively larger value than RMSLE in other ranges of r. This is presumably because the number of relatively important blocks is smaller in the road-traffic volume dataset compared to the mobility demand dataset. In the road-traffic dataset in Section 4.3, only a small number of blocks have high importance and many other blocks have low importance. In the road-traffic dataset, RMSLE has a large value especially when $r < 0.1$ because those small numbers of important blocks are dropped when $r < 0.1$. In contrast, in the mobility demand dataset, since the number of relatively important blocks is larger, RMSLE of the proposed system has a relatively large value when $r < 0.2$.

4.5 Chapter summary

To reduce the volume of transmitted data used as input for real-time spatial information prediction while maintaining the prediction accuracy, this chapter proposed an IoT device control system that uses the importance of data extracted from the machine learning model used for prediction. Importance of data is obtained by measuring how much the collected data by mobile sensors will contribute to the prediction accuracy of real-time spatial information. The proposed system extracts the importance of data by applying feature selection methods. This enables the mobile IoT devices in the proposed system to avoid transmitting less important data (in terms of how much the data contributes to the prediction accuracy) to an edge server. Performance evaluations with road-traffic and mobility-demand prediction scenarios demonstrated that the proposed system reduces the volume of data transmission for real-time predic-

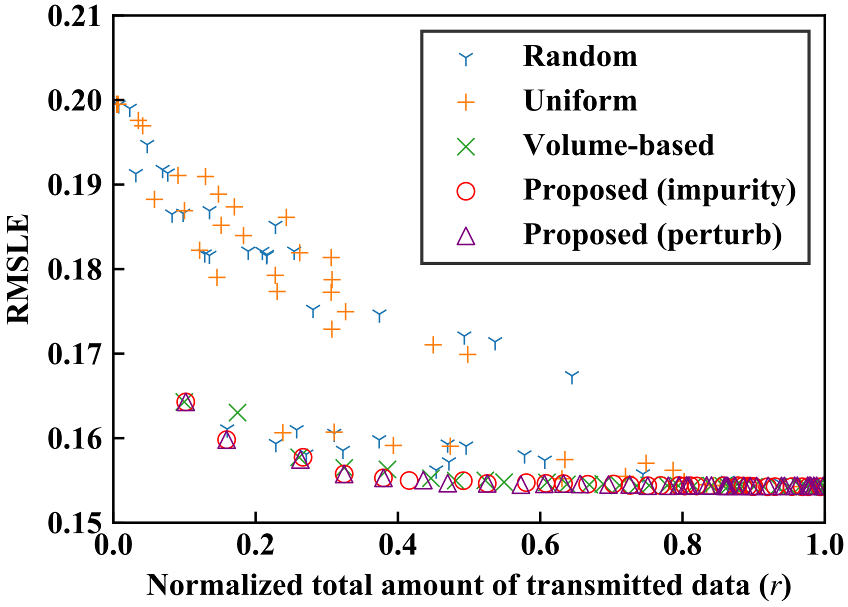


Figure 4.13: Prediction error vs. transmission rate for mobility demand prediction ($\Delta_B = 1000 \times 1000$ (m^2)). (©2019 IEEE.)

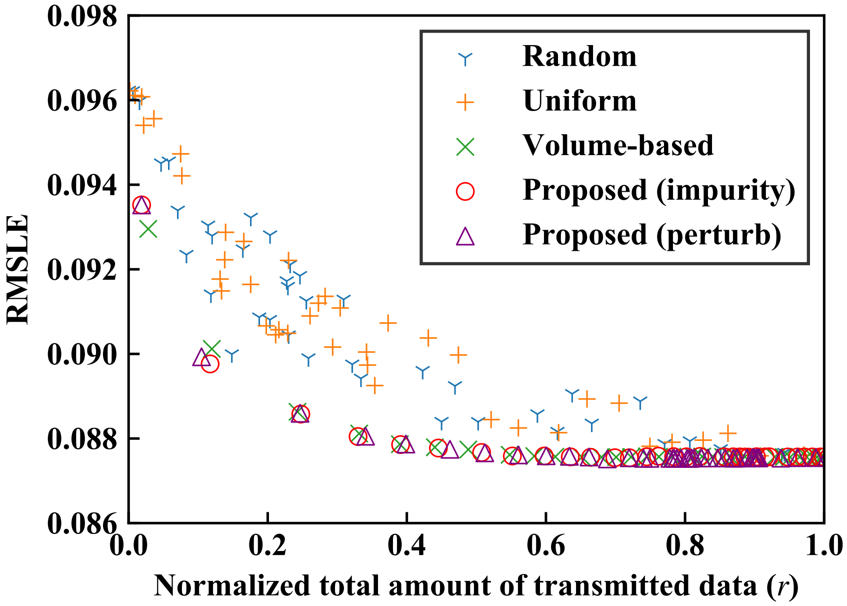


Figure 4.14: Prediction error vs. transmission rate for mobility demand prediction ($\Delta_B = 500 \times 500$ (m^2)). (©2019 IEEE.)

tion while achieving the same level of prediction accuracy as the benchmark methods.

Chapter 5

Data importance aware periodic machine learning model update for sparse mobile crowdsensing

5.1 Overview

Sparse mobile crowdsensing has been actively studied as a means of collecting data for real-time inference of spatial information [92, 93]. In sparse mobile crowdsensing for real-time inference of spatial information, first participants collect data and transmit the data to the server, and then the server aggregates the collected data and performs inference of spatial information. Participants include mobile IoT devices such as smartphones, autonomous cars, wearable devices, and unmanned aerial vehicles. Inferred spatial information can be road traffic, taxi demand, air quality or temperature. Inference algorithms include K-nearest neighbors (KNN) [94], compressive sensing [95], and machine learning algorithms such as LSTM [96] and 3D convolutional neural network (CNN) [97]. Studies on sparse mobile crowdsensing have addressed the tradeoff between the quality of resulted spatial information and sensing cost (e.g., smartphone energy consumption, network bandwidth, and incentives). These studies have achieved accurate inference of spatial information under the limited sensing cost by evaluating how valuable each data is and collecting only part of data that is evaluated to be valuable.

However, real-time inference of spatial information with sparse mobile crowdsensing using machine learning has not sufficiently considered the change of the nature of data over time. For example, when inferring a road-traffic volume map from mobility logs, newly available roads or facilities can drastically change how people make their way to their destinations. In this case, the inference model that was trained previously only knows the patterns of traffic before the new roads or facilities opened. This leads to the decreased accuracy of the resulting road-traffic volume map unless the inference model is re-trained using the newly collected mobility logs as training data.

The challenge for re-training of the inference model is to collect new training data as well as inference data efficiently. The data for re-training, as well as the data for inference, consumes sensing costs. Both data for re-training and inference should be collected efficiently. However, data that contributes to the improvement of the accuracy does not necessarily contribute to re-training model.

This chapter proposes a framework that periodically updates the model by evaluating the importance of the data in terms of both inference and re-training and giving priority in collecting important data. The proposed framework considers the importance of data for inference in conjunction with the importance of data for re-training. The importance of data for inference is a metric of how much that data will contribute to the inference accuracy. The importance of data for re-training is a metric of how much that data will contribute to re-train the inference model. The proposed framework collects important data with higher priority and periodically re-trains the inference model using the collected data. By considering both importances and collecting important data with higher priority, the proposed framework enables re-training of the inference model to follow the change of the nature of collected data and resulting spatial information, and thus improves long term accuracy of real-time inference of spatial information.

The rest of this chapter is organized as follows. Section 5.2 reviews the prior efforts on sparse mobile crowdsensing and related studies. Section 5.3 presents the details of the proposed framework and the problem formulation. Section 5.4 provides performance evaluation with practical scenarios and real-world datasets. Section 5.5 concludes this chapter.

5.2 Sparse mobile crowdsensing

Sparse mobile crowdsensing is a crowdsensing paradigm that reconstructs or predicts desired data from the data sensed in different sub-areas to reduce the required number of sensing tasks allocated, thus lowering overall sensing cost (e.g., smartphone energy consumption, network bandwidth, and incentives) while ensuring data quality. While sparse mobile crowdsensing studies achieve higher data quality with lower sensing costs, they do not sufficiently consider the change of the nature of data over time.

There have been sparse mobile crowdsensing studies that employ compressive sensing for reconstructing the desired data from sparsely collected data and active learning for choosing the best data for sensing [98–101].

5.2.1 Compressive sensing

Compressive sensing theory is a powerful tool to reconstruct a sparse vector or matrix [95]. Many compressive sensing algorithms align sensing data into a sparse spatiotemporal matrix and reconstruct the matrix by matrix completion techniques. KNN and singular value decomposition (SVD) are commonly used as matrix completion methods for compressive sensing.

KNN method is used to interpolate the missing values in the collected matrix [94]. To interpolate the missing values, KNN uses a weighted average of the values of the K nearest neighbors. When interpolating a matrix, KNN can be performed on rows or columns, which are called spatial KNN (KNN-S) or temporal KNN (KNN-T), respectively.

SVD is a standard method for matrix completion based on low-rank approximation [102]. Given a partially collected sensing matrix C, SVD reconstructs the full sensing matrix $\bar{S}$ based on the low-rank property:

$$\min \ rank(\bar{S}) \tag{5.1}$$

$$\text{s.t. } C = \bar{S} \circ B, \tag{5.2}$$

where B is a cell collection matrix that indicates whether a cell is collected or not. Directly solving this problem is hard because it is known as a nonconvex problem. A common practice for solving SVD problem is to change the rank minimization to minimizing the sum of $\boldsymbol{L}$ and $\boldsymbol{R}$'s Frobenius norms:

$$\min \lambda(||L||_F^2 + ||R||_F^2) + ||LR^T \circ B - C||_F^2, \tag{5.3}$$

where λ is used to control the trade-off between rank minimization and accuracy fitness [95]. To get the optimal $\bar{S}$, a procedure called alternate least squares can be used, which estimates $\boldsymbol{L}$ and $\boldsymbol{R}$ iteratively [103].

5.2.2 Active learning

To solve the problem of choosing the best cells for sensing, the recent techniques on active learning for matrix completion [104–106], which employ different criteria to actively choose the entry in a matrix, are all applicable.

Query by committee (QBC) [104] is one of the common active matrix completion methods. 'Committee' here refers to a set of several matrix completion algorithms. The QBC method quantifies the prediction uncertainty based on the level of disagreement among the committee members. Each matrix completion algorithms in a committee are applied to the partially collected data matrix to infer the missing values. The variance of prediction among the committee members of each missing entry is taken as a measure of uncertainty of that entry. Based on the uncertainty measured by QBC, sensing tasks for the missing entries with higher uncertainty can have higher priority.

5.3 Proposed framework

5.3.1 System model

The proposed framework performs real-time inference of spatial information from collected data while periodically updating the model by evaluating the importance of the data in terms of both inference and re-training and giving priority to collecting important data. Fig. 5.1 illustrates the system model of the proposed framework.

To formally explain and discuss the proposed framework, this work first defines several terms that describe concepts on mobile crowdsensing in this chapter. Time slot $t \in \{1, \ldots, T\}$ is the unit time in which the proposed framework performs operations such as sensing, inference, and importance calculation. Location $l \in \{1, \ldots, L\}$ is the geographical unit area where participants collect sensing data. Collected matrix C is a matrix whose rows correspond to time slots and columns correspond to locations. Each element $C_{t,l}$ records either actual data collected by a participant if the data in location l at time slot t is sensed and collected, or zero otherwise. Sensing matrix S is a matrix whose shape is the same as a collected matrix and records the ground truth data in each of its element $S_{t,l}$. Inferred sensing matrix $\bar{S}_{t,l}$ is a matrix whose shape is the same as a sensing matrix and records inferred value of $S_{t,l}$ in each of its element $\bar{S}_{t,l}$. Binary index matrix B is a matrix whose shape is the same as a sensing matrix. Each element $B_{t,l}$ indicates whether the data of location l at time slot t is collected or not. $B_{t,l} = 0$ indicates the data is missing and $B_{t,l} = 1$ indicates the data is collected. Thereby, a collected matrix C is an incomplete sensing matrix S, which can be presented using the element-wise product of S and B as:

$$C = S \circ B. \tag{5.4}$$

Cell is a spatio-temporal unit that refers to a geographical location in a specific time slot. A cell corresponds to an element in a collected matrix, sensing matrix, inferred sensing matrix, and binary index matrix. Allocating a task means the server asks a participant in a specific cell to collect data and transmit it to the server.

The objective of the proposed framework is to minimize the error between the inferred spatial information and the real spatial information with a limited number of tasks allocated to the participants. That is,

$$\min \; error(\bar{S}, S) \tag{5.5}$$

$$\text{s.t.} \; \sum_{l=1}^{L} B_{t,l} \leq N_t \; (\forall t \in \{1, \ldots, T\}), \tag{5.6}$$

$$\bar{S} = infer(C), \tag{5.7}$$

$$C = S \circ B, \tag{5.8}$$

where N_t is the limit of the number of tasks that can be allocated in time slot t and $infer(\cdot)$ is the missing data inference algorithm that calculates the

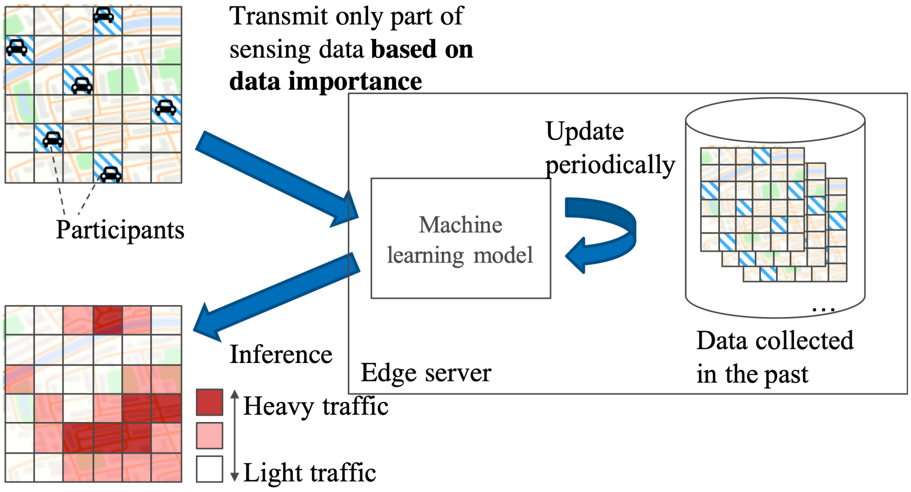


Figure 5.1: System model.

inferred sensing matrix $\bar{S}$ from the collected matrix C. The decision variable of this formula is B.

5.3.2 Control procedure

The proposed framework repeats the following four key steps: task allocation, missing data inference, inference model update, and data importance update.

Task allocation

The server evaluates the importance of each cell (t, l) and allocates tasks to the cells in descending order of the evaluated importance of cells under the sensing cost limit constraint described in (5.6).

The importance of cell (t, l), which is denoted as $I_{t,l}$, is defined as:

$$I_{t,l} = (1 - \alpha)F_{t,l} + \alpha E_{t,l}, \tag{5.9}$$

where $F_{t,l}$ denotes the importance of cell (t, l) in terms of inference, $E_{t,l}$ denotes the importance of cell (t, l) in terms of re-training, and α is a parameter that adjusts weight between the importance for inference and re-training.

$F_{t,l}$ is the inference importance of the cell (t, l). $F_{t,l}$ can be calculated from the inference model using feature selection methods [93,107]. Feature selection methods, in general, are used for selecting a set of variables (features) from the input that can efficiently describe the input data while reducing effects from noise or irrelevant variables and still provide good prediction results [34]. In this framework, feature selection methods are used to calculate inference importance from the inference model. By calculating the inference importance using feature selection methods, the proposed framework evaluates how much that data will contribute to the inference accuracy. The details of the actual feature selection methods used in performance evaluation can be found in Section 5.4.2.

$E_{t,l}$ is the re-training importance of the cell (t,l). $E_{t,l}$ is calculated by the inference error between the inferred sensing data and the ground truth sensing data. The inference error of a cell implies how much the current inference model can be potentially improved when the data of the cell is used for re-training of the inference model. By calculating the re-training importance from the inference error, the proposed framework evaluates how much the data of each cell will contribute to the re-training of the inference model.

Details on the calculation and updating process of inference importance and re-training importance are described in Section 5.3.2.

The following assumptions related to the task allocation are made to keep the fundamental part of the framework simple. First, there is a sufficient number of participants so that the server can find at least one participant to allocate a sensing task in any location at any time slot. Second, every participant can collect true sensing values when a task is allocated to her or him. Third, every sensing can be completed in enough short period of time compared to the length of a time slot. This ensures that every sensing is completed in a single time slot and every participant who is allocated a task does not move out of the current location during that time slot.

Missing data inference

The server infers the data of cells that are not allocated a task by using the inference model (Fig. 5.2).

The inference model receives the collected sensing data in the last few time slots as input X and infers the sensing data of the current time slot as

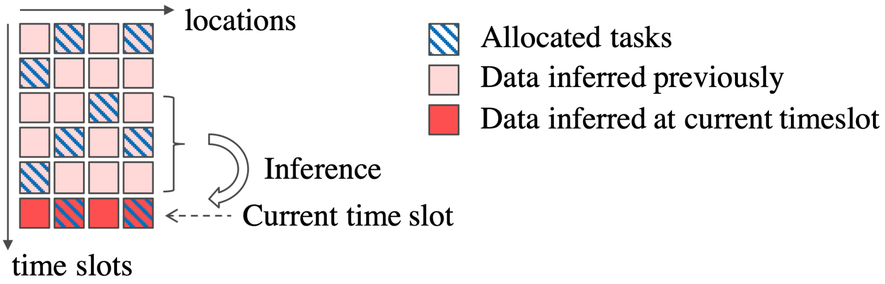


Figure 5.2: Missing data inference.

output y. X is a matrix whose shape is $T_{\text{in}} \times L$, where T_{in} denotes the number of past time slots used for inference, and L denotes the total number of the locations. Each element is a data from collected matrix: $X_{i,j} = C_{t'-T_{\text{in}}+i-1,j}$ ($\forall i \in \{1, \ldots, T_{\text{in}}\}, \forall j \in \{1, \ldots, L\}$), where t' denotes current time slot. y is a vector whose size is L. y_l is an inferred sensing data of location l in current time slot t'. The infered data $\bar{S}_{t',l}$ of a cell (t', l) is complemented with either the inferred data y_l if $B_{t',l} = 0$ (i.e. sensed data is missing) or $S_{t',l}$ otherwise.

Inference model update

The server re-trains the inference model in every T_{u} time slots, where T_{u} is the update interval of the inference model. The inference model is re-trained using the inferred data $\bar{S}$ in the last T_{r} time slots, where T_{r} is the number of time slots of data for re-training. Note that the proposed framework assumes that the re-training can be completed during a time slot.

Data importance update

The server updates the inference importance $F_{t,l}$ and the re-training importance $E_{t,l}$ in every time slot respectively.

$F_{t,l}$ is updated as:

$$F_{t,l} = feature\_importance(t, l), \tag{5.10}$$

where $feature\_importance(t, l)$ is a function that performs feature selection and calculates the inference importance of the cell (t, l). The implementation of $feature\_importance(t, l)$ depends on the type of the inference model used in the proposed framework. Some possible example implementation can be found in Section 5.4.

$E_{t,l}$ is updated as:

$$E_{t,l} = \begin{cases} error(\bar{S}_{t,l}, C_{t,l}) & (\text{if } B_{t,l} = 1) \\ E_{t-1,l} & (\text{otherwise}) \end{cases}, \tag{5.11}$$

where $error(\cdot)$ is a function that measures the error between the inferred data $\bar{S}_{t,l}$ and the collected data $C_{t,l}$. $E_{t,l}$ is updated when $B_{t,l} = 1$ because the error cannot be calculated unless the actual data $C_{t,l}$ is collected and transmitted to the server.

5.4 Performance evaluation

5.4.1 Evaluation scenario

The proposed framework is evaluated with the following two scenarios: road-traffic volume inference and taxi demand inference.

Road-traffic volume inference

In the road-traffic volume inference scenario, autonomous cars are assumed to collect the number of cars as a participant when allocated a task. The autonomous cars are assumed to be able to count the number of cars in a location at a time slot by using sensors such as onboard cameras or LiDAR sensors. The server collects the data of the number of cars from part of cells that are evaluated to be important. The server infers the missing part of the data and outputs the full sensing data of the number of cars.

San Francisco cabs dataset [90] is used for the evaluation in the road-traffic volume inference scenario. The dataset includes the location logs of 536 taxies for 25 days. In this evaluation, the number of cars in the dataset is aggregated in 20 × 20 location every 10 minutes. The size of each location is 100m × 100m. Table 5.1 shows the details on the dataset. Following parameters are used in this evaluation; $T_{\text{in}} = 6$, $T_u = 144$, and $T_r = 2016$. Figure 5.3 shows the

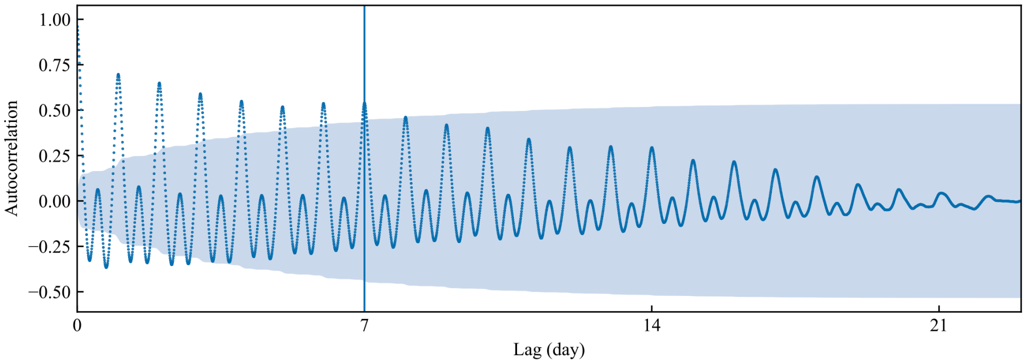


Figure 5.3: Temporal correlation of San Francisco dataset.

temporal correlation of the average value in locations of San Francisco dataset.

Taxi demand inference

In the taxi demand inference scenario, the number of pickups is predicted on the basis of people detection from sensing data collected by taxis.

Uber pickups dataset [108] is used for the evaluation in the taxi demand inference scenario. The dataset includes the location of over 4 million Uber taxi pickups for 183 days. In this evaluation, the number of cars in the dataset is aggregated in 20 × 20 location every 10 minutes. The size of each location is 100m × 100m. Table 5.1 shows the details on the dataset. Following parameters are used in this evaluation; $T_{\text{in}} = 6$, $T_u = 144$, and $T_r = 2016$. Figure 5.4 shows the temporal correlation of the average value in locations of Uber pickups dataset.

5.4.2 Implementation of proposed method

RF-I

RF-I uses random forest for missing data inference algorithm and impurity importance and inference error for optimal task allocation algorithm.

The impurity method calculates feature importance on the basis of the 'impurity' index used in decision tree models [43]. The impurity method in this

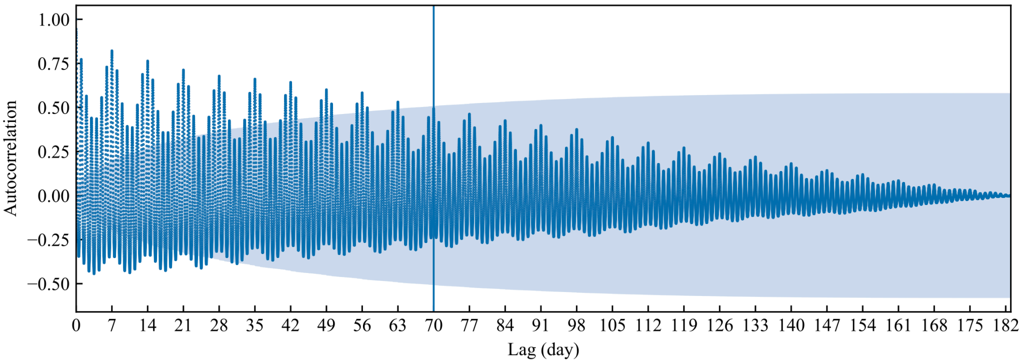


Figure 5.4: Temporal correlation of Uber pickups dataset.

Table 5.1: Datasets for performance evaluation.

	San Francisco cabs	Uber pickups
Period	2008/05/17 – 2008/06/10	2014/04/01 – 2014/09/30
Total no. of logs	11,220,505	4,534,327
Training data length	14 days	14 days
Evaluation data length	11 days	169 days
Location size	100 (m) × 100 (m)	
No. of locations	20 × 20	
Time slot length	10 (minutes)	

evaluation is implemented by the `feature_importances_` function of the random forest regressor of scikit-learn. By applying this function, the importance of each input feature, i.e., the importance of each element x_{ij}, is obtained. To simplify the evaluation, the importance of blocks is calculated by taking the sum for i. The perturb method calculates feature importance by adding noise to the subset of input features and examining the increase of error [45].

RF-P

The perturb method calculates the importance of block j by:

$$(RMSE(\hat{y}', y) - RMSE(\hat{y}, y))^2, \tag{5.12}$$

where y is the number of cars in blocks, $\hat{y}$ is the predicted value of y, and $\hat{y}'$ is the predicted value when the input values of block j in the training data are multiplied by 1.5.

5.4.3 Benchmarks

KNN-S-Q

KNN-S-Q uses KNN-S for missing data inference algorithm and QBC for optimal task allocation algorithm. The committee of QBC for KNN-S-Q includes KNN-S, KNN-T, and SVD. Details on KNN-S and QBC are described in Section 5.2.

KNN-T-Q

KNN-T-Q uses KNN-T for missing data inference algorithm and QBC for optimal task allocation algorithm. The committee of QBC for KNN-T-Q includes KNN-S, KNN-T, and SVD. Details on KNN-T and QBC are described in Section 5.2.

CS-Q

CS-Q uses SVD for missing data inference algorithm and QBC for optimal task allocation algorithm. The committee of QBC for CS-Q includes KNN-S, KNN-T, and SVD. Details on SVD and QBC are described in Section 5.2.

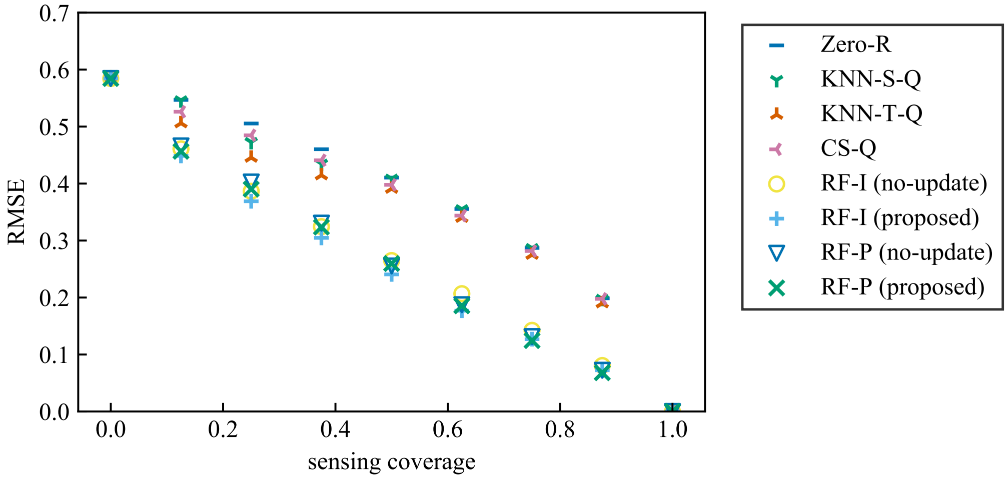


Figure 5.5: RMSE vs. sensing coverage (San Francisco dataset).

Zero-R

Zero-R uses zero filling for missing data inference algorithm and random selection for optimal task allocation algorithm. It fills non-collected cells with zero and collects data from N_t cells randomly in time slot t.

RF-I with no update, RF-P with no update

RF-I with no update and RF-P with no update are the same as RF-I and RF-P described in Section 5.4.2, respectively, except that RF-I with no update and RF-P with no update omit model update.

5.4.4 Results

Figure 5.5 shows the inference accuracy in RMSE versus the sensing coverage. Note that sensing coverage refers to the ratio of the number of allocated tasks to the number of total cells. The unit of the value of RMSE is the same as the unit of predicted values, which is the number of cars for the results on the San Francisco dataset. Zero-R, KNN-S-Q, KNN-T-Q, CS-Q, RF-I (no-update), RF-P (no-update), and the proposed framework with RF-I and RF-P are plotted. The result shows that basically, as the sensing coverage increases, RMSE

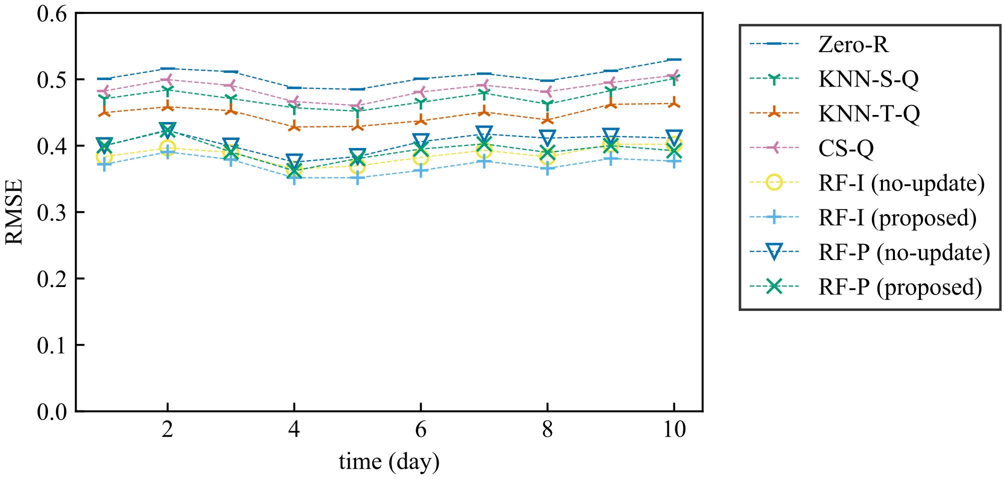


Figure 5.6: RMSE vs. time (San Francisco dataset).

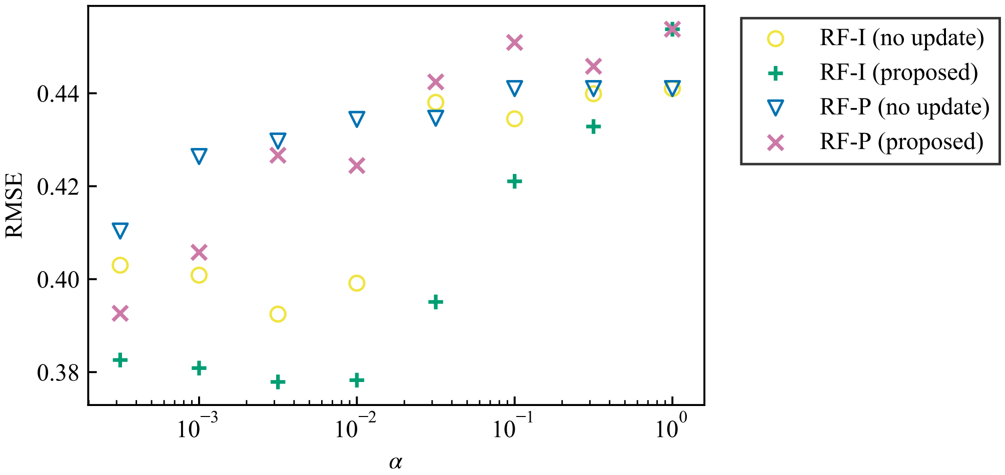


Figure 5.7: RMSE vs. α (San Francisco dataset).

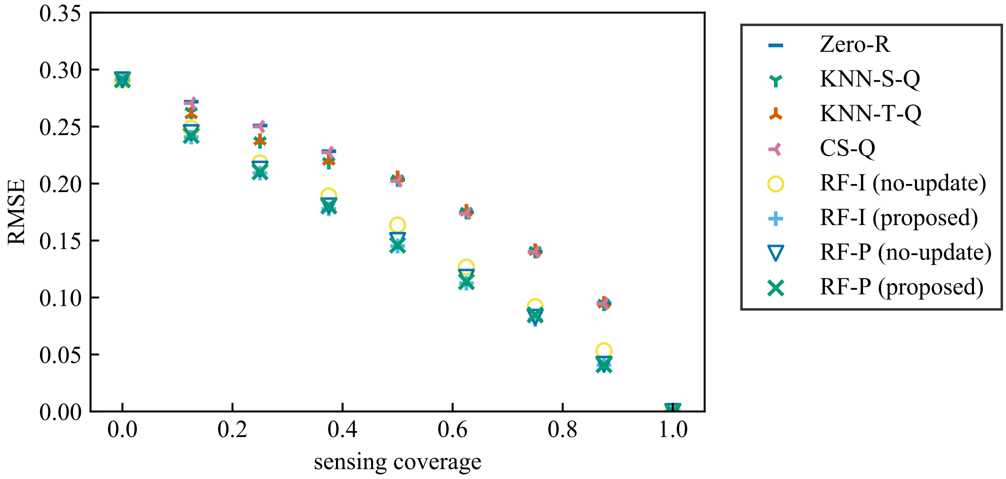


Figure 5.8: RMSE vs. sensing coverage (Uber pickups dataset).

decreases because simply, the spatial coverage of collected data contributes to the improvement of the inference accuracy. As seen in this figure, the proposed framework performed better in RMSE than the benchmarks.

Figure 5.6 shows RMSE versus time when sensing coverage is 0.25, which is an important observation because this chapter addresses periodical update of inference models. As seen in this figure, RMSE is time-varying. However, the superiority of the proposed frameworks is ensured over the time.

Figure 5.7 shows how the coefficient α affects the performance when sensing coverage is 0.25. When $\alpha = 1$, the proposed framework works in accordance with only the importance of data for re-training without considering the one for inference. On the contrary, when α is set small, the proposed framework works in accordance with only the importance of data for inference without considering the one for re-training. The proposed framework worked optimally in RMSE when α was set to 10^{-2} and RF-I is used as the feature selection method.

Figures 5.8, 5.9, and 5.10 show the results of the inference accuracy in RMSE versus the sensing coverage, RMSE versus time, and how the coefficient α affects the performance, with Uber pickups dataset, respectively. The unit of the value of RMSE is the same as the unit of predicted values, which is the number of cars for the results on the Uber pickups dataset. As observed

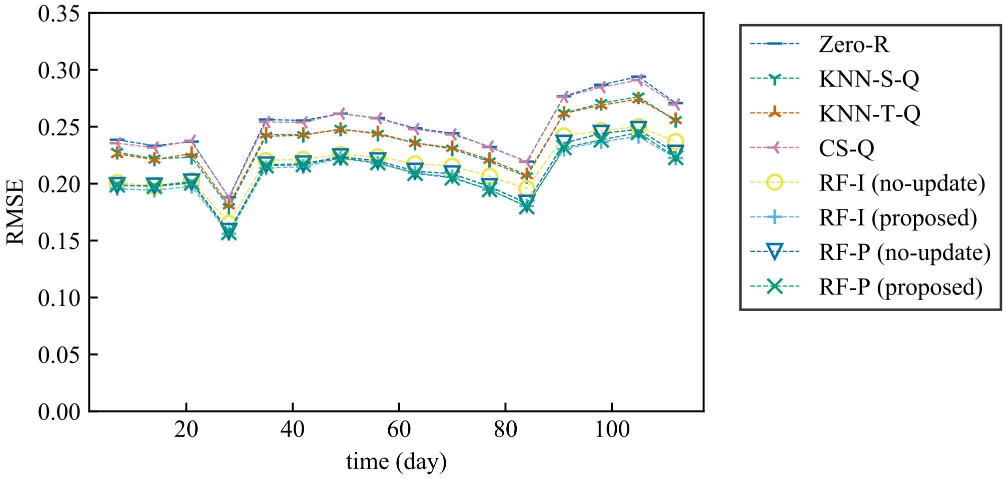


Figure 5.9: RMSE vs. time (Uber pickups dataset).

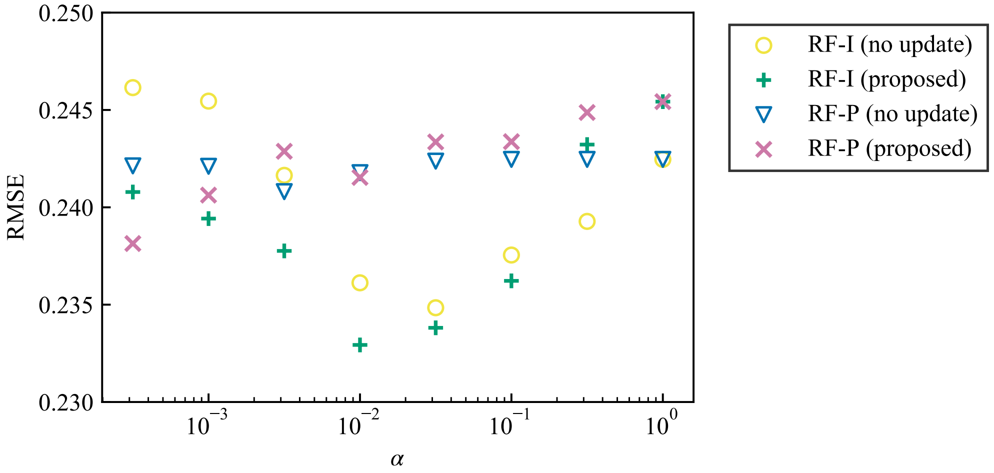


Figure 5.10: RMSE vs. α (Uber pickups dataset).

in Figs. 5.5 and 5.6, Figs. 5.8 and 5.9 show that the proposed framework performed better in RMSE than the benchmarks. Figure 5.10 shows that the proposed framework worked optimally in RMSE when α was set to 10^{-2} and RF-I is used as the feature selection method.

5.5 Chapter summary

This chapter proposed a framework that evaluates the importance of the data and gives priority in collecting important data for the periodical update of the inference model. The proposed framework collects important data with higher priority and periodically re-trains the inference model using the collected data. The numerical evaluation using two datasets validated the proposed framework; by considering both importance and collecting important data with higher priority, the proposed framework enables re-training of the inference model to follow the change of the nature of collected data and works for improving long term accuracy of real-time inference of spatial information.

Chapter 6

Conclusions

As smart city relates to many fields of services, it makes information networking for smart cities more challenges. As the information networking towards smart cities should satisfy various requirements from various services, the metrics on which the information networking based should be able to describe various requirements from various services. This thesis studied three specific problems about importance-aware information networking for smart cities.

Firstly, this thesis proposed a system for device sharing based on importance extracted from online social relationships between a device owner and user. For some types of smart city services, users provide sensing ability, computation capacity, or network connectivity of their personal devices to smart city services by sharing their devices with other users. When device owners share the limited resources on their devices, they generally want to reduce their costs when they share their devices with someone who is less socially close to them. This thesis answered a question: is there any system for device sharing that respects social relationships among device owners and users? The proposed system in this work automatically determines how much resources the user is allowed to use by acquiring and evaluating online social relationships between a device owner and user as a metric of the importance of transmitted data among devices. This work presented a prototype implementation and a large-scale simulation using a dataset of a real social network. Numerical results observed that the proposed system limits resource usage for guest users who are not close to the device owners. The resource sharing with guest users who are close to the device owners is not affected by the overhead of the authentication process in the proposed system. The results also showed that using online social relationships as the importance for humans enables information networking for device sharing.

Secondly, this thesis proposed an IoT device control system that uses the importance of data to reduce the amount of transmitted data for input of a machine learning model while maintaining the prediction accuracy. Predicting real-time spatial information from data collected by mobile IoT devices is one of the most common structures of smart city services. Mobile IoT devices for real-time spatial information prediction generate an extremely high volume of data, making it impossible to collect all of it through mobile networks. Simply reducing the volume of transmitted data does not ensure the prediction accuracy of real-time spatial information. This thesis answered a question: is there any system that reduces the amount of transmitted sensor data used as input for prediction while maintaining the inference accuracy by transmitting only an important part of sensor data that contributes to the inference accuracy? This work presented an IoT device control system that reduces the amount of transmitted data used as input for real-time prediction while maintaining prediction accuracy. In this work, the proposed system was evaluated with a real-world vehicle mobility dataset in two practical scenarios using the random forest model, an extensively used machine learning model. Numerical results via simulation indicated that the proposed system achieves the same level of prediction accuracy as benchmark methods while reducing the amount of transmitted input data for real-time prediction. The results also suggested that how much the data contributes to prediction accuracy can be used to achieve efficient information networking.

Thirdly, this thesis proposed a framework that periodically updates a machine learning model used to reconstruct the partially collected data by evaluating the importance of the data in terms of both inference and re-training and prioritizing collecting important data. Sparse mobile crowdsensing is a crowdsensing paradigm that reduces the sensing cost while ensuring data quality by collecting data sparsely and reconstructing desired data using inference algorithms, including machine learning algorithms. However, real-time inference of spatial information with sparse mobile crowdsensing has not sufficiently considered the change of the nature of data over time. As a result, the accuracy of the reconstructed data can deteriorate over time. This thesis answered a question: is there any framework that efficiently collects both the data important for re-training the inference model and the data important for inference in order to maintain long term accuracy of the inference? This work presented a framework that periodically updates a machine learning model used to reconstruct data by evaluating the importance of the data in terms of both inference and re-training and prioritizing collecting important data. Evaluation results suggested that the proposed system, which updates the model periodically, achieves better accuracy compared to benchmarks over time. The results also indicated that how much the data contributes to re-training can be used to build information networking that provides a service with long term prediction accuracy.

The proposed models and a framework studied three typical application scenarios of information networking for smart cities considering the corresponding properties, respectively. Firstly, this thesis studied altruistic device sharing based on the importance of social relationships between device owners and users. Secondly, this thesis studied an IoT device control system that evaluates the importance of data to reduce the amount of transmitted data while maintaining prediction accuracy. Thirdly, this thesis studied a framework that considers both importance for inference and importance for re-training to periodically update the inference model to maintain long term accuracy of the inference. Network operators or designers of networks for smart cities can select appropriate approaches according to the specific requirements of services to achieve efficient information networking for smart cities.

For future works, there can be three directions to extend the proposed systems. Firstly, we can consider the issues of privacy risk. The privacy issues for smart cities have been discussed since the typical services of smart cities rely on data provided by the general public. It is expected that the proposed systems can reduce privacy risk by respecting the social relationship among users or collecting only a limited part of data that is important for prediction. Secondly, we can extend this thesis to consider human-machine coexistence environments. Human-machine interaction and cooperation for smart cities have been actively studied. This thesis studied the importance for humans and machines independently. Considering importance for both humans and machines will help humans and machines coexist in networks for smart cities. Thirdly, we can extend this thesis to consider multiple services on networks for smart cities. Smart cities typically provide multiple services. This thesis studied information networking for one service at a time. Considering multiple services enables more practical information networking for smart cities.

Bibliography

[1] P. Chamoso, A. González-Briones, S. Rodríguez, and J. M. Corchado, “Tendencies of technologies and platforms in smart cities: A state-of-the-art review,” *Wireless Communications and Mobile Computing*, vol. 2018, Aug. 2018.

[2] T. M. Heng and L. Low, “The intelligent city: Singapore achieving the next lap: Practitoners forum,” *Technology Analysis & Strategic Management*, vol. 5, no. 2, pp. 187–202, Jun. 1993.

[3] Q. Li and S. Lin, “Research on digital city framework architecture,” in *2001 International Conferences on Info-Tech and Info-Net: A Key to Better Life, ICII 2001 - Proceedings*, vol. 1. Institute of Electrical and Electronics Engineers Inc., Oct. 2001, pp. 30–36.

[4] T. Ishida, H. Ishiguro, and H. Nakanishi, “Connecting digital and physical cities,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2362. Springer Verlag, Oct. 2002, pp. 246–256.

[5] R. Giffinger, C. Fertner, H. Kramar, R. Kalasek, N. Milanović, and E. Meijers, “Smart cities: ranking of European mid-sized cities,” *Digital Agenda for Europe*, p. 28, Jan. 2007.

[6] C. Harrison, B. Eckman, R. Hamilton, P. Hartswick, J. Kalagnanam, J. Paraszczak, and P. Williams, “Foundations for smarter cities,” *IBM Journal of Research and Development*, vol. 54, no. 4, pp. 1–16, Jul. 2010.

[7] G. P. Hancke, B. D. C. e. Silva, and G. P. Hancke, Jr., "The role of advanced sensing in smart cities," *Sensors*, vol. 13, no. 1, pp. 393–425, Dec. 2012.

[8] "Sustainable cities and communities – Indicators for smart cities," International Organization for Standardization, International Standard, May. 2019.

[9] "Standards support smart cities — IEC," Jul. 2019. [Online]. Available: https://www.iec.ch/node/28068

[10] "Smart cities - City service continuity against disasters - The role of the electrical supply," International Electrotechnical Commission, International Standard, Jul. 2020.

[11] Z. Sang and K. Li, "ITU-T standardisation activities on smart sustainable cities," *IET Smart Cities*, vol. 1, no. 1, pp. 3–9, Jun. 2019.

[12] "About - IEEE Smart Cities." [Online]. Available: https://smartcities.ieee.org/about

[13] "Guide for the Technology and Process Framework for Planning a Smart City," Sep. 2017. [Online]. Available: https://standards.ieee.org/project/2784.html

[14] I. Jawhar, N. Mohamed, and J. Al-Jaroodi, "Networking architectures and protocols for smart city systems," *Journal of Internet Services and Applications*, vol. 9, no. 1, pp. 1–16, Dec. 2018.

[15] J. Jin, J. Gubbi, T. Luo, and M. Palaniswami, "Network architecture and QoS issues in the internet of things for a smart city," in *2012 International Symposium on Communications and Information Technologies, ISCIT 2012*, Oct. 2012, pp. 956–961.

[16] R. Liu, W. Wu, H. Zhu, and D. Yang, "M2M-oriented QoS categorization in cellular network," in *7th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2011*, Sep. 2011.

[17] R. Shinkuma, H. Kasai, K. Yamaguchi, and O. Mayora, "Relational metric: A new metric for network service and in-network resource control," in *2012 IEEE Consumer Communications and Networking Conference (CCNC).* IEEE, Jan. 2012, pp. 352–353.

[18] R. Shinkuma, Y. Sawada, Y. Omori, K. Yamaguchi, H. Kasai, and T. Takahashi, "A socialized system for enabling the extraction of potential values from natural and social sensing," in *Modeling and processing for next-generation big-data technologies.* Springer, 2015, pp. 385–404.

[19] Z. Wang, L. Sun, X. Chen, W. Zhu, J. Liu, M. Chen, and S. Yang, "Propagation-based social-aware replication for social video contents," in *Proceedings of the 20th ACM international conference on Multimedia*, Oct. 2012, pp. 29–38.

[20] H. Hu, Y. Wen, T. Chua, J. Huang, W. Zhu, and X. Li, "Joint content replication and request routing for social video distribution over cloud CDN: A community clustering method," *IEEE transactions on circuits and systems for video technology*, vol. 26, no. 7, pp. 1320–1333, Jul. 2015.

[21] R. Shinkuma and T. Nishio, "Data assessment and prioritization in mobile networks for real-time prediction of spatial information with machine learning," in *2019 IEEE First International Workshop on Network Meets Intelligent Computations (NMIC).* IEEE, Jul. 2019, pp. 1–6.

[22] R. Shinkuma, T. Nishio, Y. Inagaki, and E. Oki, "Data assessment and prioritization in mobile networks for real-time prediction of spatial information using machine learning," *EURASIP Journal on Wireless Communications and Networking*, pp. 1–19, Jul. 2020.

[23] K. Gatsis, "Adaptive scheduling for machine learning tasks over networks," *arXiv preprint arXiv:2101.10007*, Jan. 2021.

[24] H. Rachlin and B. A. Jones, "Altruism among relatives and non-relatives," *Behavioural processes*, vol. 79, no. 2, pp. 120–123, Oct. 2008.

[25] O. Curry, S. G. Roberts, and R. I. Dunbar, “Altruism in social networks: Evidence for a ‘kinship premium’,” *British Journal of Psychology*, vol. 104, no. 2, pp. 283–295, May. 2013.

[26] O. Curry and R. Dunbar, “Altruism in networks: the effect of connections,” *Biology letters*, p. rsbl20101202, Mar. 2011.

[27] P. Branas-Garza, R. Cobo-Reyes, M. P. Espinosa, N. Jiménez, J. Kovářík, and G. Ponti, “Altruism and social integration,” *Games and Economic Behavior*, vol. 69, no. 2, pp. 249–257, Jul. 2010.

[28] F. Harrison, J. Sciberras, and R. James, “Strength of social tie predicts cooperative investment in a human social network,” *PLoS One*, vol. 6, no. 3, p. e18338, Mar. 2011.

[29] Y. Ahn, J. P. Bagrow, and S. Lehmann, “Link communities reveal multiscale complexity in networks,” *nature*, vol. 466, no. 7307, pp. 761–764, Jun. 2010.

[30] L. Lu and T. Zhou, “Link prediction in complex networks: A survey,” *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1150 – 1170, Mar. 2011.

[31] P. Jaccard, “The distribution of the flora in the alpine zone,” *New Phytologist*, vol. 11, no. 2, pp. 37–50, Feb. 1912.

[32] L. A. Adamic and E. Adar, “Friends and neighbors on the web,” *Social networks*, vol. 25, no. 3, pp. 211–230, Jul. 2003.

[33] L. Katz, “A new status index derived from sociometric analysis,” *Psychometrika*, vol. 18, no. 1, pp. 39–43, Mar. 1953.

[34] G. Chandrashekar and F. Sahin, “A survey on feature selection methods,” *Computers and Electrical Engineering*, vol. 40, no. 1, pp. 16–28, Jan. 2014.

[35] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, Mar. 2003.

[36] R. Battiti, “Using mutual information for selecting features in supervised neural net learning,” *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 537–550, Jul. 1994.

[37] G. Forman, “An extensive empirical study of feature selection metrics for text classification,” *Journal of Machine Learning Research*, vol. 3, pp. 1289–1305, Mar. 2003.

[38] P. Pudil, J. Novovičová, and J. Kittler, “Floating search methods in feature selection,” *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119–1125, Nov. 1994.

[39] D. E. Goldberg and J. H. Holland, “Genetic algorithms and machine learning,” *Machine Learning*, vol. 3, pp. 95–99, 1988.

[40] J. Kennedy, “Particle swarm optimization,” in *Encyclopedia of machine learning*. Springer, 2011, pp. 760–766.

[41] R. Kohavi and G. H. John, “Wrappers for feature subset selection,” *Artificial intelligence*, vol. 97, no. 1-2, pp. 273–324, Dec. 1997.

[42] A. L. Blum and P. Langley, “Selection of relevant features and examples in machine learning,” *Artificial intelligence*, vol. 97, no. 1-2, pp. 245–271, Dec. 1997.

[43] B. H. Menze, B. M. Kelm, R. Masuch, U. Himmelreich, P. Bachert, W. Petrich, and F. A. Hamprecht, “A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data,” *BMC Bioinformatics*, vol. 10, no. 1, p. 213, Jul. 2009.

[44] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[45] M. Gevrey, I. Dimopoulos, and S. Lek, “Review and comparison of methods to study the contribution of variables in artificial neural network models,” *Ecological Modelling*, vol. 160, no. 3, pp. 249–264, Feb. 2003.

[46] Q. Shen, R. Diao, and P. Su, "Feature selection ensemble," *Turing-100*, vol. 10, pp. 289–306, Jun. 2012.

[47] J. Sun, G. Hong, Y. Wong, M. Rahman, and Z. Wang, "Effective training data selection in tool condition monitoring system," *International Journal of Machine Tools and Manufacture*, vol. 46, no. 2, pp. 218–224, Feb. 2006.

[48] R. E. Schapire and Y. Freund, *Boosting: Foundations and algorithms.* MIT press, May. 2012.

[49] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems.* Springer, Dec. 2000, pp. 1–15.

[50] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[51] T. Teubner, "Thoughts on the sharing economy," in *Proceedings of the International Conference on e-Commerce*, vol. 11, Jul. 2014, pp. 322–326.

[52] C. J. Martin, "The sharing economy: A pathway to sustainability or a nightmarish form of neoliberal capitalism?" *Ecological Economics*, vol. 121, pp. 149–159, Jan. 2016.

[53] "Fon." [Online]. Available: https://fon.com/

[54] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, Oct. 2009.

[55] T. Nishio, R. Shinkuma, T. Takahashi, and N. B. Mandayam, "Service-oriented heterogeneous resource sharing for optimizing service latency in mobile cloud," in *Proceedings of the first international workshop on Mobile cloud computing & networking.* ACM, Jul. 2013, pp. 19–26.

[56] A. Kansal, S. Nath, J. Liu, and F. Zhao, "SenseWeb: An infrastructure for shared sensing," *IEEE MultiMedia*, vol. 14, no. 4, pp. 8–13, Oct. 2007.

[57] J. Beal, K. Usbeck, J. Loyall, and J. Metzler, "Opportunistic sharing of airborne sensors," in *Distributed Computing in Sensor Systems (DCOSS), 2016 International Conference on.* IEEE, May. 2016, pp. 25–32.

[58] J. Beal, K. Usbeck, J. Loyall, M. Rowe, and J. Metzler, "Adaptive task reallocation for airborne sensor sharing," in *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W)*, Sep. 2016, pp. 168–173.

[59] T. Zhu, Y. Gu, T. He, and Z. Zhang, "eShare: A capacitor-driven energy storage and sharing network for long-term operation," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems.* ACM, Nov. 2010, pp. 239–252.

[60] N. Eagle and A. Pentland, "Reality mining: Sensing complex social systems," *Personal and Ubiquitous Computing*, vol. 10, no. 4, pp. 255–268, May. 2006.

[61] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE Rap: Social-based forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1576–1589, Nov. 2011.

[62] M. Yamada, R. Shinkuma, and T. Takahashi, "Cooperative networking in heterogeneous infrastructure multihop mobile networks," in *2006 IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications*, Sep. 2006, pp. 1–5.

[63] D. Zhang, R. Shinkuma, and N. B. Mandayam, "Bandwidth exchange: An energy conserving incentive mechanism for cooperation," *IEEE Transactions on Wireless Communications*, vol. 9, no. 6, pp. 2055–2065, Jun. 2010.

[64] P. Shankar, B. Nath, L. Iftode, V. Ananthanarayanan, and L. Han, "Sbone: Personal device sharing using social networks," *Rutgers University, Tech. Rep. DCS-TR-666*, 2010.

[65] X. Wang, M. Chen, T. Kwon, L. Jin, and V. C. M. Leung, "Mobile traffic offloading by exploiting social network services and leveraging opportunistic device-to-device sharing," *IEEE Wireless Communications*, vol. 21, no. 3, pp. 28–36, Jun. 2014.

[66] X. Wang and V. C. Leung, "SNS-based mobile traffic offloading by opportunistic device-to-device sharing," *The Future of Wireless Networks: Architectures, Protocols, and Services*, vol. 21, p. 327, 2015.

[67] X. Wang, Z. Sheng, S. Yang, and V. C. M. Leung, "Tag-assisted social-aware opportunistic device-to-device sharing for traffic offloading in mobile social networks," *IEEE Wireless Communications*, vol. 23, no. 4, pp. 60–67, Aug. 2016.

[68] L. Garton, C. Haythornthwaite, and B. Wellman, "Studying online social networks," vol. 3, Jun. 1997.

[69] "Bumbee Labs - Framtidens besöksflöden är här." [Online]. Available: https://www.bumbeelabs.com/en

[70] T. Kaya and H. Bicen, "The effects of social media on students' behaviors; Facebook as a case study," *Computers in Human Behavior*, vol. 59, pp. 374–379, Jun. 2016.

[71] D. Hardt, "The OAuth 2.0 authorization framework," Oct. 2012.

[72] H. Annuar, B. Shanmugam, A. Ahmad, N. B. Idris, S. H. AlBakri, and G. N. Samy, "Enhancement and implementation of network access control architecture for virtualization environments," in *2013 International Conference on Informatics and Creative Multimedia*, Sep. 2013, pp. 314–320.

[73] "SNAP: Network datasets: Brightkite." [Online]. Available: https://snap.stanford.edu/data/loc-brightkite.html

[74] E. Cho, S. A. Myers, and J. Leskovec, “Friendship and mobility: User movement in location-based social networks,” pp. 1082–1090, Aug. 2011.

[75] World Health Organization, *Global status report on road safety*, Dec. 2015.

[76] Centre for Economics and Business Research, “The future economic and environmental costs of gridlock in 2030,” *Report for INRIX*, Jul. 2014.

[77] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.

[78] X. Wang, W. Wu, and D. Qi, “Mobility-aware participant recruitment for vehicle-based mobile crowdsensing,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4415–4426, May. 2018.

[79] Z. He, J. Cao, and X. Liu, “High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility,” in *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, Apr. 2015, pp. 2542–2550.

[80] H. G. Seif and X. Hu, “Autonomous driving in the iCity-HD maps as a key challenge of the automotive industry,” *Engineering*, vol. 2, no. 2, pp. 159–162, Jun. 2016.

[81] Future Market Insights, “Cyber-physical system market: Will china be able to surpass western europe in terms of growth in the coming years: Global industry analysis (2013 - 2017) & opportunity assessment (2018 - 2028),” Apr. 2018.

[82] V. Cao, K. Chu, N. Le-Khac, M. Kechadi, D. Laefer, and L. Truong-Hong, “Toward a new approach for massive lidar data processing,” in *2015 2nd IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services (ICSDM)*, Jul. 2015, pp. 135–140.

[83] A. Manjeshwar and D. P. Agrawal, "TEEN: A routing protocol for enhanced efficiency in wireless sensor networks," in *Proceedings 15th International Parallel and Distributed Processing Symposium. IPDPS 2001*, Apr. 2001, pp. 2009–2015.

[84] Y. Ma, Y. Guo, X. Tian, and M. Ghanem, "Distributed clustering-based aggregation algorithm for spatial correlated sensor networks," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 641–648, Mar. 2011.

[85] J. Yue, W. Zhang, W. Xiao, D. Tang, and J. Tang, "Energy efficient and balanced cluster-based data aggregation algorithm for wireless sensor networks," *Procedia Engineering*, vol. 29, pp. 2009–2015, Jan. 2012.

[86] A. Sinha and D. K. Lobiyal, "Performance evaluation of data aggregation for cluster-based wireless sensor network," *Human-centric Computing and Information Sciences*, vol. 3, no. 1, p. 13, Aug. 2013.

[87] I. Czarnowski, "Distributed learning with data reduction," in *Transactions on Computational Collective Intelligence IV*, N. T. Nguyen, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 3–121.

[88] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning.* MIT press, Oct. 2016.

[89] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, Oct. 2011.

[90] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "CRAWDAD dataset epfl/mobility (v. 2009-02-24)," Downloaded from https://crawdad.org/epfl/mobility/20090224/cab, Feb. 2009.

[91] M. Zeng, T. Yu, X. Wang, V. Su, L. T. Nguyen, and O. J. Mengshoel, "Improving demand prediction in bike sharing system by learning global features," in *KDD 2016*, Aug. 2016.

[92] L. Wang, D. Zhang, Y. Wang, C. Chen, X. Han, and A. M'Hamed, "Sparse mobile crowdsensing: Challenges and opportunities," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 161–167, Jul. 2016.

[93] Y. Inagaki, R. Shinkuma, T. Sato, and E. Oki, "Prioritization of mobile IoT data transmission based on data importance extracted from machine learning model," *IEEE Access*, vol. 7, pp. 93 611–93 620, Jul. 2019.

[94] T. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[95] L. Wang, D. Zhang, A. Pathak, C. Chen, H. Xiong, D. Yang, and Y. Wang, "CCS-TA: Quality-guaranteed online task allocation in compressive crowdsensing," *ACM Ubicomp*, pp. 683–694, Sep. 2015.

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

[97] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[98] G. Quer, R. Masiero, G. Pillonetto, M. Rossi, and M. Zorzi, "Sensing, compression, and recovery for WSNs: Sparse signal modeling and monitoring framework," *IEEE Transactions on Wireless Communications*, vol. 11, no. 10, pp. 3447–3461, Aug. 2012.

[99] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang, "A compressive sensing approach to urban traffic estimation with probe vehicles," *IEEE Transactions on Mobile Computing*, vol. 12, no. 11, pp. 2289–2302, Nov. 2013.

[100] L. Kong, M. Xia, X. Y. Liu, G. Chen, Y. Gu, M. Y. Wu, and X. Liu, "Data loss and reconstruction in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 11, pp. 2818–2828, Nov. 2014.

[101] L. Xu, X. Hao, N. D. Lane, X. Liu, and T. Moscibroda, "More with less: Lowering user burden in mobile crowdsourcing through compressive

sensing," in *UbiComp 2015 - Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing.* New York, New York, USA: Association for Computing Machinery, Inc, Sep. 2015, pp. 659–670.

[102] T. Hastie, R. Tibshirani, G. Sherlock, M. Eisen, P. Brown, and D. Botstein, "Imputing missing data for gene expression arrays," *Technical Report, Division of Biostatistics, Stanford University*, Sep. 1999.

[103] L. Kong, M. Xia, X. Liu, M. Wu, and X. Liu, "Data loss and reconstruction in sensor networks," *2013 Proceedings IEEE INFOCOM*, pp. 1654–1662, Apr. 2013.

[104] S. Chakraborty, J. Zhou, V. Balasubramanian, S. Panchanathan, I. Davidson, and J. Ye, "Active matrix completion," *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 81–90, Dec. 2013.

[105] R. Karimi, C. Freudenthaler, A. Nanopoulos, and L. Schmidt-Thieme, "Non-myopic active learning for recommender systems based on matrix factorization," *Proceedings of the 2011 IEEE International Conference on Information Reuse and Integration, IRI 2011*, pp. 299–303, Aug. 2011.

[106] D. J. Sutherland, B. Póczos, and J. Schneider, "Active learning and search on low-rank matrices," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. Part F128815, pp. 212–220, Aug. 2013.

[107] Y. Yamada, R. Shinkuma, T. Sato, and E. Oki, "Feature-selection based data prioritization in mobile traffic prediction using machine learning," in *2018 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2018, pp. 1–6.

[108] "Uber pickups in New York city," Downloaded from https://www.kaggle.com/fivethirtyeight/uber-pickups-in-new-york-city, Nov. 2016.

Publication List

Journal Papers

1. R. Shinkuma, T. Nishio, **Y. Inagaki**, and E. Oki, “Data assessment and prioritization in mobile networks for real-time prediction of spatial information using machine learning,” *EURASIP Journal on Wireless Communications and Networking*, May. 2020.

2. **Y. Inagaki**, R. Shinkuma, T. Sato, and E. Oki, “Prioritization of mobile IoT data transmission based on data importance extracted from machine learning model,” *IEEE Access*, vol. 7, pp. 93611 - 93620, Jul. 2019.

3. **Y. Inagaki** and R. Shinkuma, “Shared-resource management using online social-relationship metric for altruistic device sharing,” *IEEE Access*, vol. 6, pp. 23191 - 23201, Apr. 2018.

4. R. Shinkuma, Y. Sugimoto, and **Y. Inagaki**, “Weighted network graph for interpersonal communication with temporal regularity,” *Springer Soft Computing*, Nov. 2017.

International Conference Papers

1. D. Masuda, R. Shinkuma, **Y. Inagaki**, and E. Oki, “Blockchain framework for realtime streaming data generated in image sensor networks for smart monitoring,” *2nd conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS2020)*, Sep. 2020.

2. R. Shinkuma, R. Takagi, **Y. Inagaki**, E. Oki, and F. Xhafa, "Incentive mechanism for mobile crowdsensing in spatial information prediction using machine learning," *AINA2020*, Apr. 2020.

3. **Y. Inagaki** and R. Shinkuma, "Authentication control system for mobile device sharing based on online social relationships," *2017 IEEE Global Communications Conference (GLOBECOM 2017)*, Dec. 2017.

Technical Reports and Local Conference Papers

1. **Y. Inagaki**, R. Shinkuma, T. Sato, and E. Oki, "Prioritized transmission of mobile IoT data using machine learning models," 電子情報通信学会, 超知性ネットワーキングに関する分野横断型研究会 (RISING), Nov. 2019.

2. **Y. Inagaki** and R. Shinkuma, "デバイス共有のためのオンライン上の社会的関係性に基づいた認証制御の実証," 信学技報, vol. 117, no. 71, MoNA2017-5, pp. 225-228, Jun. 2017.

3. **Y. Inagaki** and R. Shinkuma, "B-15-17 オンライン上の社会的関係性に基づいたデバイス共有の認証制御方式の検討," 電子情報通信学会総合大会講演論文集, Mar. 2017.

Awards

1. Best Poster Presentation Award at 電子情報通信学会, 超知性ネットワーキングに関する分野横断型研究会 (RISING), Nov. 2019.

2. 独立行政法人情報処理推進機構 (IPA) 2018 年度未踏 IT 人材発掘・育成事業 採択, Apr. 2018.