High-Confidence Computing 2 (2022) 100045

Contents lists available at ScienceDirect

High-Confidence Computing

homepage: www.elsevier.com/locate/hcc

Edge computing-Based mobile object tracking in internet of things

Yalong Wu[a], Pu Tian[b], Yuwei Cao[c], Linqiang Ge[d], Wei Yu[b,*]

[a] Dept. of Computer Science and Engineering, North Central College, Naperville, IL 60540Paraguay
[b] Dept. of Computer and Information Sciences, Towson University, Towson, MD, 21252
[c] Dept. of Computer Science, University of Illinois at Chicago, Chicago, IL 60607United States
[d] School of Computer Science, Columbus State University, Columbus, GA 31907

ARTICLE INFO

Keywords:
Internet of things
Edge computing
Architecture
Mobile object tracking
Vector auto regression

ABSTRACT

Mobile object tracking, which has broad applications, utilizes a large number of Internet of Things (IoT) devices to identify, record, and share the trajectory information of physical objects. Nonetheless, IoT devices are energy constrained and not feasible for deploying advanced tracking techniques due to significant computing requirements. To address these issues, in this paper, we develop an edge computing-based multivariate time series (EC-MTS) framework to accurately track mobile objects and exploit edge computing to offload its intensive computation tasks. Specifically, EC-MTS leverages statistical technique (i.e., vector auto regression (VAR)) to conduct arbitrary historical object trajectory data revisit and fit a best-effort trajectory model for accurate mobile object location prediction. Our framework offers the benefit of offloading computation intensive tasks from IoT devices by using edge computing infrastructure. We have validated the efficacy of EC-MTS and our experimental results demonstrate that EC-MTS framework could significantly improve mobile object tracking efficacy in terms of trajectory goodness-of-fit and location prediction accuracy of mobile objects. In addition, we extend our proposed EC-MTS framework to conduct multiple objects tracking in IoT systems.

1. Introduction

With the advance of Internet of Things (IoT) and big data sharing and analytics, massive number of IoT devices (sensors, actuators, etc.) are deployed to enable the monitoring and control of things under minor or no human intervention [1–7]. Mobile object tracking, a typical IoT application, along with other smart-world applications, such as smart grid, smart transportation, smart health, and smart manufacturing, are involving more and more IoT devices so that automatic monitoring and tracking on physical objects, including moving targets, vehicles, assets, etc., can be supported [8–12].

Nonetheless, it is a common practice that IoT devices are energy and computing constrained, and they are not competent to consistently handle complex mobile object tracking tasks, which are computationally intensive and consume lots of energy resources. Thus, computation offloading in IoT systems is a critical issue for accurate and efficient mobile object tracking. Our proposed scheme in this paper, designated Edge Computing-based Multivariate Time Series (EC-MTS) framework, endeavors to apply complex tracking technique to improve mobile object tracking performance in IoT systems and employs edge computing to offload computation intensive tasks that energy constrained IoT devices cannot handle. With an intent to understand how to track mobile objects in IoT systems and design corresponding solutions, a number of research efforts have been conducted [13–19]. The existing efforts demonstrate that the advance of mobile object tracking could support a diverse array of smart-world IoT applications (e.g., video surveillance, robot navigation, etc.) in dynamic environments.

Nonetheless, most of the existing efforts merely focused on enhancing mobile object tracking accuracy through the adoption of rather complex tracking algorithms (distributed searching, correlation filtering, etc.) [20] and overlooked the energy constraint of IoT devices in running computation intensive techniques. Consequently, the battery of IoT devices could be quickly drained due to computation intensive tracking tasks, and eventually incur failures on delivering sustainable mobile object tracking services. Thus, it is important to develop a mobile object tracking framework in IoT systems, which can ensure a high-level object tracking accuracy, and does not burden energy constrained IoT devices in dealing with computation intensive tasks at the same time.

The primary contributions of this paper are summarized as follows:

- **Framework:** We propose an edge computing-based multivariate time series (EC-MTS) framework to model mobile object trajectory over a long time duration in IoT systems and predict mobile object locations at multiple upcoming time points. EC-MTS is designed to improve the goodness-of-fit in trajectory modeling and the accu-

* Corresponding author.
E-mail addresses: ywu1601@noctrl.edu (Y. Wu), ptian1@students.towson.edu (P. Tian), ycao43@uic.edu (Y. Cao), ge_linqiang@columbusstate.edu (L. Ge), wyu@towson.edu (W. Yu).

https://doi.org/10.1016/j.hcc.2021.100045
Received 27 May 2021; Received in revised form 15 September 2021; Accepted 15 September 2021

racy in mobile object location prediction. Specifically, EC-MTS utilizes the statistical technique vector auto regression (VAR) [21,22] to model mobile object trajectory with the consideration of the historical mobile object location data. Particularly, EC-MTS takes advantage of edge computing to offload the intensive computation tasks, and could thus walk back arbitrary steps to thoroughly revisit trajectory data. Given the understanding of the mobile object trajectory, EC-MTS finally predicts the mobile object locations at prospective time points on the basis of the multivariate least squares estimators (MLSE) [23,24]. With the arbitrary historical trajectory investigation, EC-MTS could obtain desirable trajectory goodness-of-fit and object location prediction accuracy for mobile object tracking. We also extend the EC-MTS framework from the perspective of multiple objects tracking in IoT systems.

- **Evaluation:** We evaluate our proposed EC-MTS framework through extensive performance evaluation in terms of trajectory goodness-of-fit and mobile object location prediction accuracy. To be specific, we conduct the experimentation from the perspectives of data collection and tracking evaluation. In data collection, we assemble real-world object location data to constitute the mobile object trajectory. In tracking evaluation, our proposed EC-MTS framework is assessed, consisting of trajectory goodness-of-fit evaluation and mobile object location prediction evaluation. Our experimental results confirm that our EC-MTS achieves desirable performance with respect to trajectory goodness-of-fit and location prediction accuracy.

The remainder of this paper is organized as follows: In Section 2, we introduce system model. In Section 3, we present our approach in detail. In Section 4, we show experimental results to validate the efficacy of our approach. In Section 5, we discuss an extension. In Section 6, we review related works. Finally, we conclude the paper in Section 7.

2. System model

Mobile object tracking [13–15] is designed to detect moving targets, keep an eye on their trajectories, and send their location reports to interested parties. Thus, it highly relies on the sensing and computing capabilities of IoT devices, which hold the functionalities of object detecting, signal processing, resource scheduling, and location prediction, among others. IoT systems involve a tremendous number of devices (sensors, actuators, etc.) and enable connectivity between them with little or no human intervention. It has been the backbone of smart-world systems, supporting a wide range of mobile object tracking applications (e.g., criminal chasing in the public surveillance and vehicle tracing on the highway and others).

To effectively track mobile objects, IoT devices need to support most of the mobile object tracking functionalities, such as object sensing and location prediction, which consume a lot of energy resources. Thus, some IoT devices could fail quickly due to energy drain. For example, object location prediction algorithms are critical for mobile object tracking, and they could enable the actuators, one type of IoT devices, in IoT systems to activate devices around the mobile object and deactivate those that are far away. Nonetheless, complicated mobile object prediction schemes commonly have very high computing complexity, which could cost the lifetime of IoT devices.

With the development of modern technology, IoT systems have evolved into three domains: device domain, network domain, and computing domain [2,12], as shown in Fig. 1. In mobile object tracking, IoT devices (i.e., sensors) in the device domain are used to detect the moving targets and collect their location data. The location information is forwarded through the network domain to the centralized cloud domain that could be far away from IoT devices. The cloud domain has enormous computing resources, and is capable of hosting the computation intensive functionalities (e.g., location prediction). Finally, the processed useful information is forwarded back to IoT devices for further actions.

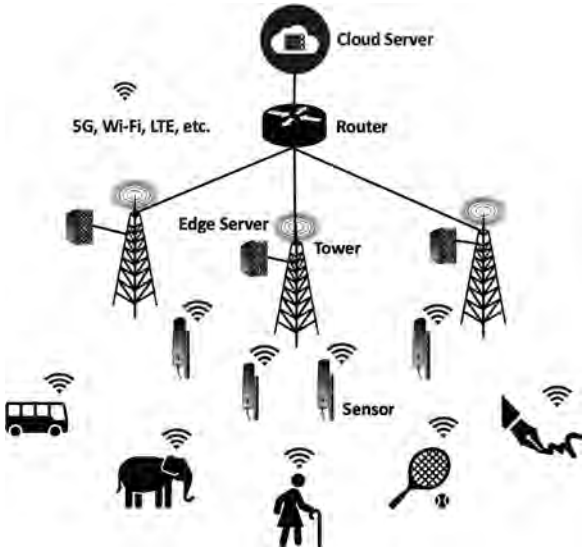


Fig. 1. EC-MTS Platform.

Nonetheless, the long latency of data transmission from IoT devices to the centralized cloud can not only pose significant overhead to the network, leading to network congestion, but also might eventually lose the track of moving targets. Thus, we involve edge computing servers at network edge, which are more close to IoT devices. The edge computing servers could provide sufficient computing resources for the computation intensive functionalities in mobile object tracking tasks, while reducing the data transmission cost to the centralized cloud. With the edge computing infrastructure, our proposed EC-MTS framework can be deployed for carrying out both object trajectory modeling and object location prediction on edge computing servers. For the sake of simplicity, we focus on EC-MTS over one mobile object, which can be extended to multiple mobile object tracking scenarios as discussed in Section 5.

3. Our approach

In this section, we first introduce the basic idea of our approach, and then illustrate its key components.

3.1. Basic idea

Our proposed EC-MTS framework tends to precisely model and predict the object trajectory so that efficient and accurate mobile object tracking in IoT systems can be provided. It utilizes the advanced statistical technique and can efficiently avoid using the energy in IoT devices.

To be specific, EC-MTS applies statistical technique called VAR (vector auto regression) [21,22] to walk back arbitrary steps, and look at historical mobile object location data to provide inference on current object location estimation. Unlike univariate auto regression time series models, VAR is a multivariate model that correlates a current variable observation with its past observations and the past observations of other variables in the system. In our case, it considers both the influence of lagged object locations on the current object location and the influence of any unobservable errors on the current object location, which will be discussed shortly. Once a mobile object trajectory is formed, EC-MTS further utilizes it to accurately predict the impending mobile object locations on the basis of multivariate least squares estimators (MLSE). In addition, EC-MTS exploits edge computing to reduce the energy burden from the computation intensive tasks and the lifetime of IoT devices is thus prolonged.

We consider two metrics, trajectory goodness-of-fit and mobile object location prediction accuracy, to assess the efficacy of EC-MTS framework. With the introduction of historical mobile object location data and

hidden errors, EC-MTS endeavors to improve the mobile object tracking efficiency in IoT systems without bundling energy constrained IoT devices.

Our proposed EC-MTS framework consists of three key components: (i) **Trajectory modeling** estimates the current mobile object location based on the historical object locations and any unobservable errors, (ii) **Trajectory parameter estimation** specifies the values for the unknown parameters in the mobile object trajectory model on the basis of multivariate least squares (MLS) technique [23,24], and (iii) **Object location prediction** forecasts multiple upcoming mobile object locations by leveraging the MLSE. In the following, we describe these components in detail.

3.2. Trajectory modeling

Trajectory modeling is critical to mobile object tracking as it could help us understand the characteristics of the objects' motion trail, and thus provide useful guidance for mobile object location prediction. In this paper, we consider the locations of mobile objects as three dimensional coordinates and associate them with time stamps. Thus, a sequential number of mobile objects' locations will be a multivariate time series from the perspective of statistics [25]. Denote the location of a mobile object at time point t ($t = 0, 1, 2, \ldots$) as $\boldsymbol{\mathcal{L}}_t$, then $\boldsymbol{\mathcal{L}}_t$ could be represented as a 3×1 (3 rows and 1 column) vector $(x_t, y_t, z_t)'$. If we apply the statistical technique VAR(p) (p is a positive integer and $p \leq t$) to estimate $\boldsymbol{\mathcal{L}}_t$, we have

$$\boldsymbol{\mathcal{L}}_t = \mathbf{c} + \boldsymbol{\mathcal{A}}_1 \boldsymbol{\mathcal{L}}_{t-1} + \boldsymbol{\mathcal{A}}_2 \boldsymbol{\mathcal{L}}_{t-2} + \ldots + \boldsymbol{\mathcal{A}}_p \boldsymbol{\mathcal{L}}_{t-p} + \mathbf{e}_t. \tag{1}$$

Here, $\mathbf{c}$ is a constant matrix to smooth the equation, $\boldsymbol{\mathcal{A}}$ are the coefficient matrices for historical mobile object locations $\boldsymbol{\mathcal{L}}_{t-1}$ through $\boldsymbol{\mathcal{L}}_{t-p}$, and $\mathbf{e}_t$ is the unobservable error matrix at time point t. Note that the matrices $\mathbf{c}$, $\boldsymbol{\mathcal{L}}$, $\mathbf{e}_t$ in Equation (1) are in the format of 3×1 and the matrices $\boldsymbol{\mathcal{A}}$ are in the format of 3×3. With Equation (1), we shall also notice that the statistical technique VAR(p) estimates the current location of the mobile object by walking back p steps to include the influences of the historical mobile object locations ($\boldsymbol{\mathcal{L}}_{t-1}, \boldsymbol{\mathcal{L}}_{t-2}, \ldots, \boldsymbol{\mathcal{L}}_{t-p}$) on the current one ($\boldsymbol{\mathcal{L}}_t$), VAR($p$) also looks at the random errors at the current time point, which might affect the current location of the mobile object. This location estimation procedure could indicate the mobile object trajectory well. If we unfold Equation (1) by using $(x_t, y_t, z_t)'$, we have

$$\begin{aligned}\begin{pmatrix} x_t \\ y_t \\ z_t \end{pmatrix} &= \begin{pmatrix} c^1 \\ c^2 \\ c^3 \end{pmatrix} + \begin{pmatrix} \mathcal{A}_1^{11} & \mathcal{A}_1^{12} & \mathcal{A}_1^{13} \\ \mathcal{A}_1^{21} & \mathcal{A}_1^{22} & \mathcal{A}_1^{23} \\ \mathcal{A}_1^{31} & \mathcal{A}_1^{32} & \mathcal{A}_1^{33} \end{pmatrix} \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \end{pmatrix} + \ldots \\ &+ \begin{pmatrix} \mathcal{A}_p^{11} & \mathcal{A}_p^{12} & \mathcal{A}_p^{13} \\ \mathcal{A}_p^{21} & \mathcal{A}_p^{22} & \mathcal{A}_p^{23} \\ \mathcal{A}_p^{31} & \mathcal{A}_p^{32} & \mathcal{A}_p^{33} \end{pmatrix} \begin{pmatrix} x_{t-p} \\ y_{t-p} \\ z_{t-p} \end{pmatrix} + \begin{pmatrix} e_t^1 \\ e_t^2 \\ e_t^3 \end{pmatrix}, \end{aligned} \tag{2}$$

where x_t could be explicitly denoted as follows:

$$\begin{aligned} x_t &= c^1 + \mathcal{A}_1^{11} x_{t-1} + \mathcal{A}_1^{12} y_{t-1} + \mathcal{A}_1^{13} z_{t-1} + \ldots \\ &+ \mathcal{A}_p^{11} x_{t-p} + \mathcal{A}_p^{12} y_{t-p} + \mathcal{A}_p^{13} z_{t-p} + e_t^1. \end{aligned} \tag{3}$$

Note that the coordinate x_t in Equation (3) is estimated not only by its own lagged values (i.e., x_{t-1},..., x_{t-p}), but also by the lagged values (i.e., y_{t-1},..., y_{t-p} and z_{t-1},..., z_{t-p}) of other coordinates y_t and z_t, respectively. The unobservable error term e_t^1 is additionally included. Likewise, both y_t and z_t could be derived from Equation (2) and they are affected by the other coordinates and the random errors as well. Thus, VAR(p) is a comprehensive statistical technique, which is capable of capturing the dependencies between diverse coordinates of multiple location time series. This helps build an accurate mobile object trajectory.

3.3. Trajectory parameter estimation

With the description in Section 3.2, we understand that the current mobile object location could be estimated on the basis of the historical object location data. Nonetheless, the parameter set $\boldsymbol{\mathcal{P}} = (\mathbf{c}\boldsymbol{\mathcal{A}}_1\boldsymbol{\mathcal{A}}_2 \cdots \boldsymbol{\mathcal{A}}_p)$ in the VAR(p) model (i.e., Equation (1) in Section 3.2) are all unknown. Thus, it is a necessity to estimate the most proper parameter values for $\boldsymbol{\mathcal{P}}$ based on the mobile object location observations. Then, the fitted VAR(p) model could be utilized for accurate object location prediction. For the sake of simplicity, we first transform the VAR(p) model into a concise form

$$\boldsymbol{\mathcal{L}}_t = \boldsymbol{\mathcal{P}} \boldsymbol{\mathcal{Z}}_{t-1} + \mathbf{e}_t, \tag{4}$$

where $\boldsymbol{\mathcal{L}}_t = (\boldsymbol{\mathcal{L}}_p \boldsymbol{\mathcal{L}}_{p+1} \cdots \boldsymbol{\mathcal{L}}_t)$ and $\mathbf{e}_t = (\mathbf{e}_p\ \mathbf{e}_{p+1} \cdots \mathbf{e}_t)$. Particularly, $\boldsymbol{\mathcal{Z}}$ is expressed as

$$\boldsymbol{\mathcal{Z}}_{t-1} = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ \boldsymbol{\mathcal{L}}_{p-1} & \boldsymbol{\mathcal{L}}_p & \ldots & \boldsymbol{\mathcal{L}}_{t-1} \\ \boldsymbol{\mathcal{L}}_{p-2} & \boldsymbol{\mathcal{L}}_{p-1} & \ldots & \boldsymbol{\mathcal{L}}_{t-2} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\mathcal{L}}_0 & \boldsymbol{\mathcal{L}}_1 & \ldots & \boldsymbol{\mathcal{L}}_{t-p} \end{pmatrix}, \tag{5}$$

please refer to [23,24] for detailed explanations.

In this paper, we utilize statistical technique multivariate least squares (MLS) technique to estimate the most appropriate values for the parameter set $\boldsymbol{\mathcal{P}}$, which is able to make VAR(p) to ultimately approximate the real mobile object trajectory. Specifically, MLS tends to choose a parameter set that minimizes the sum of the squares of the differences between the estimated and observed mobile object location values. Denote the number of the observed mobile object locations as τ (a positive integer). We could represent the squares function $S(\boldsymbol{\mathcal{P}})$ as

$$S(\boldsymbol{\mathcal{P}}) = \|\boldsymbol{\mathcal{L}}_\tau - \boldsymbol{\mathcal{P}} \boldsymbol{\mathcal{Z}}_{\tau-1}\|^2, \tag{6}$$

this would give us the estimated parameter set $\hat{\boldsymbol{\mathcal{P}}}$ as

$$\hat{\boldsymbol{\mathcal{P}}} = \arg \min_{\boldsymbol{\mathcal{P}}} S(\boldsymbol{\mathcal{P}}) = \boldsymbol{\mathcal{L}}_\tau \boldsymbol{\mathcal{Z}}_{\tau-1}\prime(\boldsymbol{\mathcal{Z}}_{\tau-1}\boldsymbol{\mathcal{Z}}_{\tau-1}\prime)^{-1}, \tag{7}$$

and $\hat{\boldsymbol{\mathcal{P}}}$ is the multivariate least squares estimators (MLSE) and they could minimize the random errors that are incurred by the difference between the estimated and observed mobile object locations. Thus, MLS is a best-effort technique to model the real mobile object trajectory.

3.4. Object location prediction

Mobile object location prediction is critical to conducting accurate object tracking in IoT systems. The accurately predicted object locations could actually provide guidance for activating the nearest IoT devices around the mobile object. With the elaboration in Section 3.3, we understand that the fitted VAR(p) model for mobile object trajectory is $\boldsymbol{\mathcal{L}}_t = \hat{\boldsymbol{\mathcal{P}}} \boldsymbol{\mathcal{Z}}_{t-1} + \mathbf{e}_t$ under τ observations. Thus, it is straight forward to obtain the one step ahead prediction $\tilde{\boldsymbol{\mathcal{L}}}_{\tau+1}$ as follows

$$\tilde{\boldsymbol{\mathcal{L}}}_{\tau+1} = \hat{\boldsymbol{\mathcal{P}}} \boldsymbol{\mathcal{Z}}_\tau + \mathbf{e}_\tau, \tag{8}$$

and we could utilize the predicted $\tilde{\boldsymbol{\mathcal{L}}}_{\tau+1}$ as the mobile object location at time point $\tau + 1$ to subsequently carry out multi-step ahead predictions.

It is worth noting that the energy consumption of VAR(p) in our EC-MTS framework will be larger when the value of p (the number of backward steps to investigate historical mobile object location data) increases. In other words, when p increases, more historical mobile object location data is included in the whole procedure of trajectory modeling, trajectory parameter estimation, and object location prediction. Nonetheless, we offload this intensive computation task to the edge computing infrastructure by leveraging our proposed EC-MTS platform in Section 2. Thus, the energy constraint of IoT devices is properly overcome. Moreover, VAR in statistics prefers low order parameter, and it is a good practice to limit the value of p up to 12, because the presence of variations most likely will occur within 12 steps (say for monthly data due to seasonality [21]). We also conduct traversal of p up to 12 in our experimentation.

4. Performance evaluation

We now demonstrate the experimental results of our proposed EC-MTS framework with respect to trajectory goodness-of-fit and mobile object location prediction accuracy. We have conducted our extensive simulation in Python.

Data Collection. In this particular experimentation, we choose the global positioning system (GPS) tracking data to validate the efficacy of our proposed EC-MTS framework. In the GPS tracking system, the latitude, longitude, and altitude of the mobile object are continuously recorded with GPS loggers and GPS phones, which will then be transmitted to data centers for analysis. Specifically, we have utilized the user GPS tracking data collected from the city of Beijing, China, and it is publicly available at **GeoLife GPS Trajectories**[1] [26–28]. This dataset consists of the trajectory data of 182 users, and it was recorded in a period of five years from April 2007 to August 2012. In particular, this dataset has a wide variety of trajectories associated with the users activities, including daily commute, working, traveling, sports activities, shopping, dining, etc. In each record of the dataset, there are always values for the latitude, longitude, altitude along with timestamps. The whole dataset covers a total distance of 1,292,951 km within a total of 50,176 hours.

As stated in Section 3, three dimensional data is desired in trajectory modelling, trajectory parameter estimation, and object location prediction. Thus, we extract the latitude, longitude, and altitude columns for one user from the dataset and fit them into our proposed EC-MTS framework in our experimentation. Particularly, we use the trajectory data of user 000 on 2009-04-04 and select one trajectory from each period of the day (morning: 10:09:59–10:21:54, afternoon: 14:27:49–14:35:09, and late night: 01:06:15–01:24:35) for diversity. There are 147, 102, and 248 records in those trajectories, respectively. 80% (117, 82, and 198) of them will be used for mobile object trajectory fitting, and the rest 20% (30, 20, and 50) are for mobile object location prediction.

Goodness-of-fit Evaluation. In order to obtain accurate location prediction for mobile object tracking, we first need to derive an object trajectory model that could describe the object moving phenomenon. In this experimentation, we use the following metrics to evaluate the mobile object trajectory goodness-of-fit: AIC (Akaike's information criterion), BIC (Bayesian information criterion), and HQIC (Hannan-Quinn information criterion). Specifically, AIC indicates the quality of a fitted model (the lower, the better); BIC (the lower, the better) is closely relevant to AIC, but the value of BIC may be slightly higher when more parameters are involved in the fitted model; HQIC also has the same function of AIC with lower values preferred and they tend to be close if models are well fitted. In our experimentation, we implement VAR(p) (vector auto regression) in Python, walk through the p values from 1 to 12 and choose one best-effort model to conduct mobile object location prediction.

Figs. 2 (a), 2(b), and 2(c) summarize our experimental results for VAR model selection in EC-MTS framework. Based on the evaluation results of Fig. 2(a), we understand that the AIC, BIC, and HQIC values are the smallest when p equals to 3. Thus, $p = 3$ is the best fitted model for morning trajectory data and we select it in this experimentation for morning trajectory fitting and prediction. Likewise, $p = 2$ will be utilized for both afternoon and late night trajectory fitting and prediction on the basis of the evaluation results in Figs. 2(b) and 2(c)

It is worth noting that the difference between the AIC, BIC, and HQIC values of the optimal p and those of the non-optimal ps in Fig. 2 is rather small. Thus, we would expect the performance of diverse models to differ not that much. Nonetheless, it is not desirable to choose the models with higher AIC, BIC, and HQIC values since lower values are preferred for those metrics. In this experimentation, we select a single model to conduct mobile object trajectory fitting and prediction, and p value minimizing the AIC, BIC, and HQIC values is still the optimal choice. In the future, if we decide to choose multiple models to apply model averaging, the non-optimal ps of AIC, BIC, and HQIC values close to those of the optimal p would be combined and integrated with the optimal model.

In order to further validate the goodness-of-fit of VAR(3), VAR(2), and VAR(2) in EC-MTS framework for morning, afternoon, and late night trajectories, we also look at the residual autocorrelation after their fitting. With the illustration in Section 3.2, we understand that each variable (i.e., x, y, z) in EC-MTS is affected by its own lagged values and the lagged values of the other two. Figs. 3(a) through 3(i) show the autocorrelation of moving users' latitude, longitude, and altitude (i.e., x, y and z in EC-MTS framework) residuals over the lagged values. For example, Fig. 3(a) indicates the autocorrelation between the morning latitude residuals after the latitude values are fitted by VAR(3) over the lagged morning latitude, longitude and altitude values. As we can see from the figures, the autocorrelation falls in between the shaded areas, which infers that the autocorrelation between latitude residuals is not significant after fitting and VAR(3) is a very well fitted model for morning latitude in EC-MTS. Fig. 3(b) demonstrates the longitude residual autocorrelation after longitude values are fitted by using VAR(3) over the lagged latitude, longitude, and altitude values, while Fig. 3(c) is for the altitude. They both as well indicate that VAR(3) is a well fitted model. Likewise, Figs. 3(d), 3(e), 3(f) and Figs. 3(g), 3(h), 3(i) show that VAR(2) is the well fitted model for both afternoon and late night trajectory data. It is worth noting that the thresholds for the shaded areas in Figs. 3(a) through 3(i) are set as 0.2, which is an acceptable rate of false positive. This is conventionally and universally adopted in statistics [21,22,25].

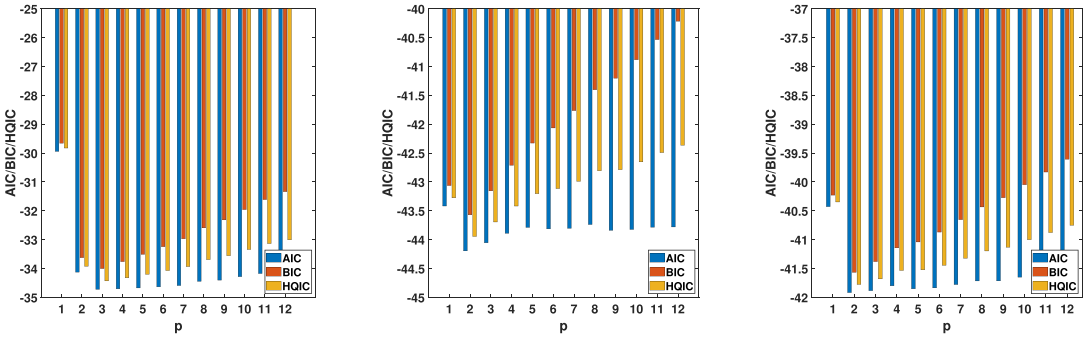
In addition, Figs. 4(a) through 4(i) demonstrate the comparison between the fitted latitude, longitude, and altitude values and their observed ones for morning, afternoon, and late night trajectories, respectively. As we can see from the figures, the actual and fitted values are nearly overlapped with each other. Thus, VAR(3) and VAR(2) are again proved to be very good for mobile object trajectory fitting.

Prediction Accuracy Evaluation. With the positive evaluation results from mobile object trajectory goodness-of-fit of VAR(3) for morning trajectory data and VAR(2) for afternoon and late night trajectory data in EC-MTS, we could now make predictions for the future mobile object locations. Specifically, we forecast ahead 30, 20, and 50 steps for morning, afternoon, and late night trajectories and compare the predicted object locations with the observed ones. Due to the space limitation, we only show the first six comparison data for morning, afternoon, and late night trajectories, respectively and the observation from the whole comparison data set is consistent. For example, Table 1 shows the comparison between the predicted latitude, longitude, and altitude values and the actual observed ones for morning trajectory. As we can see from the table, both the latitude and longitude predictions capture the trends of the observed values, and the MSE (mean squared error: 0.000430773439554 for morning latitude and 0.000225591483211 for morning longitude) of prediction is very close to 0. Thus, both the latitude and longitude predictions for morning trajectory are convincing. Nonetheless, the result for the altitude predictions shows fluctuation. This is actually reasonable and explainable, because there may be sudden changes in the height when users are going up and down in elevators, overpasses, etc. Thus, VAR(3) in EC-MTS is able to obtain accurate mobile object location prediction for morning trajectory. Likewise, Tables 2 and 3 demonstrate that VAR(2) performs well in mobile object location prediction for afternoon and late night trajectories, respectively.

5. Extension

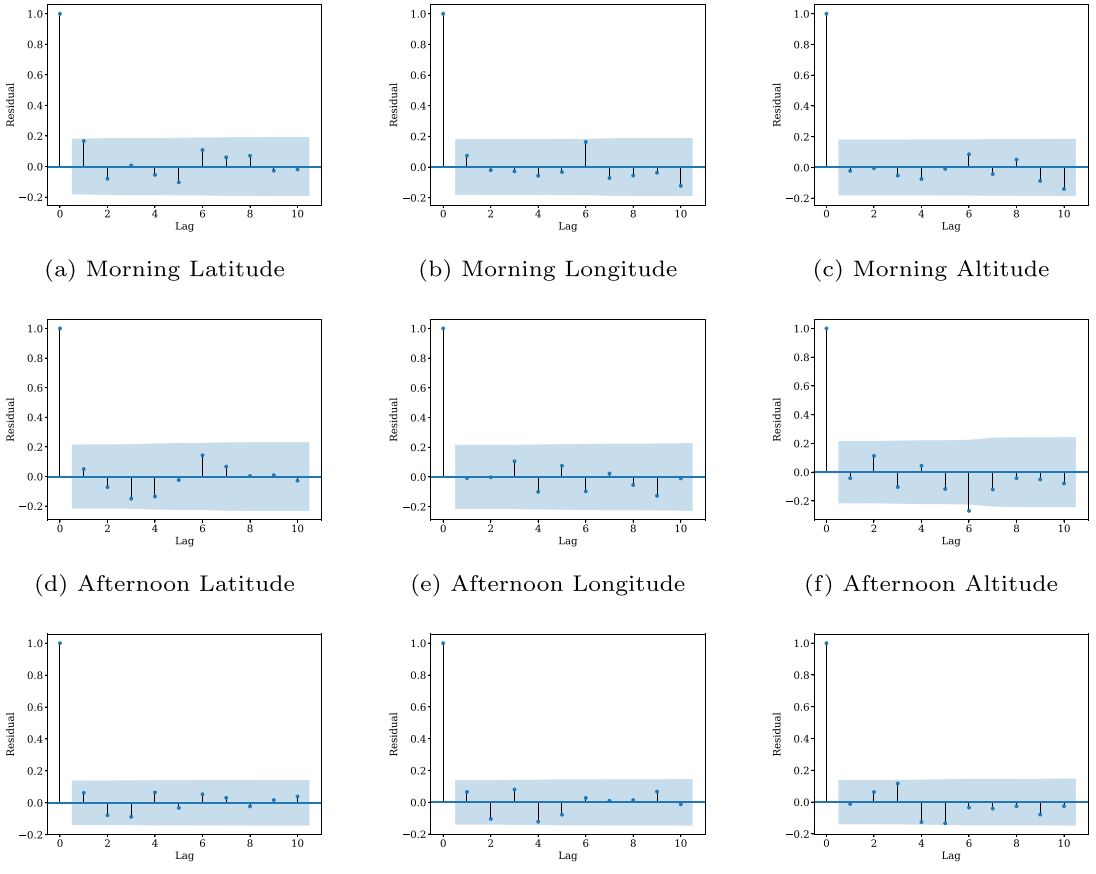
In this section, we extend our proposed EC-MTS framework to carry out the tracking of multiple objects in IoT systems. In single object tracking, we process the measurements of IoT device (sensor, camera, etc.) to determine the location of a moving object (e.g., athletes on court and cars on street), while the number of mobile objects and their locations

[1] https://www.microsoft.com/en-us/download/details.aspx?id=52367



(a) Morning (b) Afternoon (c) Late Night

Fig. 2. Trajectory Fitting Evaluation.



(a) Morning Latitude (b) Morning Longitude (c) Morning Altitude

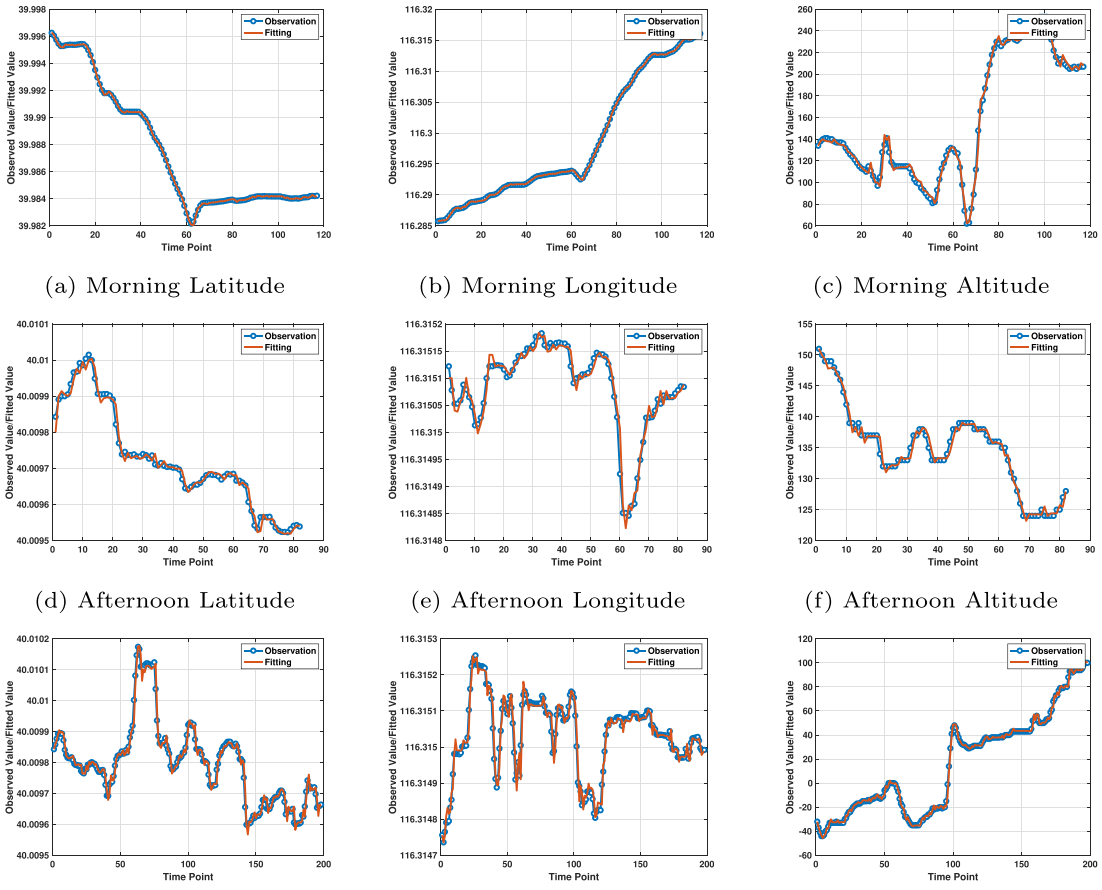
(d) Afternoon Latitude (e) Afternoon Longitude (f) Afternoon Altitude

(g) Late Night Latitude (h) Late Night Longitude (i) Late Night Altitude

Fig. 3. Autocorrelation of Trajectory Residuals.

Table 1
Morning Trajectory Prediction Evaluation.

Latitude: Observation vs. Prediction —— MSE: 0.000430773439554	
39.984202 39.984108 39.984131	40.00954219 40.00934515 40.00900074
39.984179 39.984005 39.983759	40.00853639 40.00742469 40.00685885
Longitude: Observation vs. Prediction —— MSE: 0.000225591483211	
116.316481 116.316933 116.317537	116.31511833 116.31492436 116.31459963
116.318170 116.318517 116.318504	116.31414705 116.31287534 116.31207979
Altitude: Observation vs. Prediction —— MSE: 27117.753291455003	
208.000000 201.000000 201.000000	142.51170039 152.82097859 164.28603840
199.000000 167.000000 156.000000	174.47273894 184.47318551 182.87566252



(a) Morning Latitude (b) Morning Longitude (c) Morning Altitude

(d) Afternoon Latitude (e) Afternoon Longitude (f) Afternoon Altitude

(g) Late Night Latitude (h) Late Night Longitude (i) Late Night Altitude

Fig. 4. EC-MTS Fitting with Trajectory.

Table 2
Afternoon Trajectory Prediction Evaluation.

Latitude: Observation vs. Prediction —— MSE: 5.610408135110442e-09	
40.009542 40.009542 40.009544	40.00961492 40.00962773 40.00963888
40.009567 40.009595 40.009598	40.00964889 40.00966614 40.00967347
Longitude: Observation vs. Prediction —— MSE: 1.0101777230082494e-08	
116.315085 116.315085 116.315087	116.31519294 116.31518841 116.31517523
116.315116 116.315153 116.315156	116.31515792 116.31512159 116.31510544
Altitude: Observation vs. Prediction —— MSE: 21.507589175517786	
128.000000 128.000000 128.000000	136.56639051 137.06975004 137.55180958
129.000000 135.000000 135.000000	137.96908244 138.44937863 138.48204944

Table 3
Late Night Trajectory Prediction Evaluation.

Latitude: Observation vs. Prediction —— MSE: 9.11823059895649e-09	
40.009672 40.009702 40.009683	40.00961035 40.00961399 40.00961328
40.009731 40.009698 40.009698	40.00961038 40.00960299 40.00959978
Longitude: Observation vs. Prediction —— MSE: 3.07024547517837e-08	
116.315015 116.315071 116.315123	116.31518544 116.31516802 116.31513916
116.315127 116.315191 116.315191	116.31510550 116.31503971 116.31501174
Altitude: Observation vs. Prediction —— MSE: 622.1220770880293	
103.000000 107.000000 104.000000	136.71135434 137.10246600 137.19986903
109.000000 107.000000 107.000000	137.05784651 136.30314498 135.80098490

have to be identified simultaneously in multiple objects tracking. This enables a diverse array of critical applications, such as public surveillance, automatic driving, pedestrian detection, among others [29,30]. Thus, it is important to extend our proposed EC-MTS framework in the field of multiple object tracking.

With the elaboration in Section 2, we know that the location information of mobile objects is collected by IoT devices and forwarded to edge computing severs for object trajectory modeling and object location prediction. For simplicity, we assume that the location information assembled by an IoT device could be transmitted to edge computing servers either directly (edge computing servers are within transmission range of IoT device) or indirectly (edge computing servers are out of transmission range of IoT device) via relay of other IoT devices. In the following, we discuss multiple object tracking in general without specifying any particular time points for simplicity of illustration. Nonetheless, the algorithm could be easily unfolded over time series for any mobile objects, please refer to Section 3 for details. In addition, we define the capacity of an edge computing server as the number of resource blocks it holds. It is also worth noting that the locations of edge computing servers and IoT devices do not change over time.

If we assume the number of edge computing servers in the area of interest as N_s ($N_s = 1, 2, \ldots$), then the location and capacity of the i^{th} ($i = 1, 2, \ldots, N_s$) edge computing server s_i can be represented as (x_{s_i}, y_{s_i}, z_{s_i}) and C_{s_i} (positive integer number of resource blocks), respectively. Likewise, we denote the number of IoT devices within the same area as N_d ($N_d = 1, 2, \ldots$) and the location of the j^{th} ($j = 1, 2, \ldots, N_d$) IoT device d_j as (x_{d_j}, y_{d_j}, z_{d_j}), respectively. We also denote the sensing range of d_j as R_{d_j} (positive real number). If the distance between the mobile object and one IoT device is not greater than the sensing range of the IoT device, then the IoT device can be used to collect the location information of the object. Suppose that there are N_o ($N_o = 1, 2, \ldots$) mobile objects in this area, the set of usable sensing IoT devices S_{o_k} for k^{th} ($k = 1, 2, \ldots, N_o$) mobile object o_k can be represented as follows:

$$S_{o_k} = \{d_j | \forall j \sqrt{\sum_{v \in \{x,y,z\}} (v_{d_j} - v_{o_k})^2} \leq R_{d_j}\} \tag{9}$$

where $v_{o_k} = (x_{o_k}, y_{o_k}, z_{o_k})$ is the location of the mobile object o_k.

In this extension, we suppose that one IoT device is needed to sense the arrival of an mobile object and collect its location information for processing at the edge computing servers. The status of an IoT device could be busy with sensing and transmitting information or idle without handling anything. In addition, the mobile object can be immediately detected the moment when it enters the sensing range of an IoT device. In other words, the mobile object can be detected and handled by the idle IoT device with the largest sensing range. Thus, the mobile object o_k can be first detected by the idle IoT device, which is within the set of usable sensing IoT devices S_{o_k} and most distant from o_k at the same time. Denote the status of IoT device d_j as $\mathbb{S}_{d_j}$, if we suppose $\mathbb{S}_{d_j} = 1$ when d_j is busy, then $\mathbb{S}_{d_j} = 0$ when d_j is idle. Note that there might be multiple IoT devices that are idle and have the same distance from the mobile object. Thus, the final candid set of IoT devices $\hat{S}_{o_k}$ to sense mobile object o_k can be represented by

$$\hat{S}_{o_k} = \{d_j | \forall d_j \ (d_j \in S_{o_k} \wedge \mathbb{S}_{d_j} = 0 \ \wedge \arg \max_{d_j} \sqrt{\sum_{v \in \{x,y,z\}} (v_{d_j} - v_{o_k})^2})\}. \tag{10}$$

If $\hat{S}_{o_k}$ is a non-empty set, one of the IoT devices will be randomly selected to collect the location information of mobile object o_k. Otherwise, all the IoT devices are currently busy and $\hat{S}_{o_k}$ will be recalculated as o_k moves in the area over time.

In Section 3.4, we highlight that the energy consumption of VAR(p) (vector auto regression) technique to model and predict mobile object trajectory at edge computing server in EC-MTS framework is proportional to p, which is the number of steps for VAR(p) to go back and investigate the historical mobile object location data. Assume that $\hat{d}_{o_k}$ is the chosen IoT device from Equation (10) to collect and forward the location information of o_k to edge computing server for mobile object trajectory modeling and prediction. In order to reduce transmission time, $\hat{d}_{o_k}$ will select the closest edge computing server with enough number of available resource blocks to conduct trajectory modeling and prediction for o_k. If we assume the number of resource blocks needed for VAR (p) technique when $p = 1$ as N_r and the number of backward steps to investigate the historical location data of mobile object o_k as p_{o_k}, then the number of resource blocks to model and predict the trajectory of o_k will be $cp_{o_k}N_r$, where c is a coefficient to adjust and smooth the formula.

Note that there also might be several edge computing servers that have enough available resource blocks and the same distance from the chosen IoT device $\hat{d}_{o_k}$. If we suppose the available number of resource blocks at edge computing server s_i as $C\prime_{s_i}$, the usable set of edge computing servers to handle mobile object o_k via $\hat{d}_{o_k}$ can be represented by

$$S_{\hat{d}_{o_k}} = \{s_i | \forall i \ (cp_{o_k}N_r \leq C\prime_{s_i} \wedge \arg \min_{s_i} \sqrt{\sum_{v \in \{x,y,z\}} (v_{s_i} - v_{\hat{d}_{o_k}})^2})\}. \tag{11}$$

If $S_{\hat{d}_{o_k}}$ is a non-empty set, one of the edge computing servers will be randomly utilized to conduct trajectory modeling and prediction for mobile object o_k by using VAR(p_{o_k}). Otherwise, all edge computing servers in this area are currently not available and $S_{\hat{d}_{o_k}}$ will be calculated again as o_k moves over time.

Finally, it is intuitive that the total requested number of resource blocks from diverse mobile objects in the area towards one particular edge computing server cannot exceed its capacity. Otherwise, we might risk losing the target due to resource scarcity. It is also better to make sure that the number of mobile objects in the area of interest is less than that of IoT devices. This (meaning $N_o \leq N_d$) assures that there is at least one IoT device to deal with the sensing and transmitting task of any mobile object. If we assume $\hat{s}_{\hat{d}_{o_k}}$ as the selected edge server to model and predict the trajectory of mobile object o_k, the requested resource blocks from mobile object o_k must be less than or equal to the available resource blocks at edge computing server $\hat{s}_{\hat{d}_{o_k}}$:

$$cp_{o_k}N_r \leq C\prime_{\hat{s}_{\hat{d}_{o_k}}}, \tag{12}$$

which also guarantees that the total requested number of resource blocks from all mobile objects in the area of interest is less than or equal to the total capacity of all edge computing servers in this area. This can be represented by

$$\sum_{k=1}^{N_o} cp_{o_k}N_r \leq \sum_{i=1}^{N_s} C_{s_i}. \tag{13}$$

6. Related works

The demand of computing power and storage space draws challenges for massive and accurate mobile object tracking. A number of research efforts have been conducted in the IoT context to improve the performance of mobile object tracking [13,17,31–40]. For example, Noguchi et al. [13] proposed a search architecture that utilizes real-time “live” video data from shared devices to discover the most appropriate devices for an IoT service (i.e., mobile object tracking). They also developed a distributed and dynamic search function architecture to cope with arbitrary searches. Zhang et al. [17] proposed a region proposal correlation filter fitting algorithm for edge devices in IoT. Their developed algorithm employs response confidence level to detect tracking status and update tracking model with a lightweight computing, such that it could track target under challenging conditions with high tracking accuracy and robustness. Han et al. [35] developed an effective hierarchical location caching scheme in an IoT system, which acclimates the existing location caching scheme to a hierarchical architecture of location databases to fast lookup an object and reduce the signalling traffic. Wei et al. [36] utilized the highly-directional 60 GHz millimeter-wave radios technology to enable a practical design mTrack, which leverages a

discrete beam scanning scheme to identify the initial location of object and track its trajectory. mTrack could also suppress interference from background reflections and achieve passive writing object tracking with 90-percentile error under 8 mm. In addition, Jiang et al. [37] designed a flexible framework Remix to accept a latency budget and derive an image partition and model execution plan. This plan applies off-the-shelf neural networks on non-uniformly partitioned image blocks to improve object detection accuracy by 18% to 70%.

Moreover, edge computing or fog computing is widely recognized as a promising solution for the time-sensitive and mission-critical application of object tracking [14,41–45]. For example, Gu et al. [14] developed a collaborative edge-cloud architecture to enhance object tracking in IoT. Their designed architecture featured offloading computations to the centralized cloud and utilizing convolution neural networks to regularly check edge device statuses, such that the track errors could be rectified quickly and accurately. In addition, Pudasaini et al. [45] designed a framework that detects and tracks the object in the edge devices while performing pattern recognition in the cloud device. Their framework converts the video data into text on the edge so as to reduce the traffic transmitted to the network.

7. Conclusion

In this paper, we proposed an edge computing-based multivariate time series (EC-MTS) framework for accurate mobile object tracking in IoT systems. Specifically, EC-MTS utilizes classic statistical technique VAR (vector auto regression) to model the mobile object trajectory by walking back arbitrary steps to investigate the historical object location data. Our proposed framework is also able to take advantage of the fitted best-effort object trajectory model to accurately predict mobile object locations. EC-MTS additionally exploits edge computing to offload computation intensive tasks from the energy constrained IoT devices. Our experimental results demonstrate that EC-MTS could obtain better object trajectory goodness-of-fit and object location prediction accuracy for mobile object tracking. We also extended our proposed EC-MTS framework to enable multiple objects tracking in IoT systems.

References

[1] S. Chen, R. Ma, H.-H. Chen, H. Zhang, W. Meng, J. Liu, Machine-to-machine communications in ultra-dense networksa survey, IEEE Communications Surveys & Tutorials 19 (3) (2017) 1478–1503.

[2] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, W. Zhao, A survey on internet of things: architecture, enabling technologies, security and privacy, and applications, IEEE Internet Things J. 4 (5) (2017) 1125–1142.

[3] Z. Cai, T. Shi, Distributed query processing in the edge assisted iot data monitoring system, IEEE Internet Things J. (2020).

[4] F. Liang, W. Yu, D. An, Q. Yang, X. Fu, W. Zhao, A survey on big data market: pricing, trading and protection, IEEE Access 6 (2018) 15132–15154.

[5] Z. Cai, Z. He, Trading private range counting over big iot data, in: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), IEEE, 2019, pp. 144–153.

[6] Q. Yang, L. Chang, W. Yu, On false data injection attacks against kalman filtering in power system dynamic state estimation, Security and Communication Networks 9 (9) (2016) 833–849.

[7] F. Liang, C. Qian, W.G. Hatcher, W. Yu, Search engine for the internet of things: lessons from web search, vision, and opportunities, IEEE Access 7 (2019) 104673–104691.

[8] B.R. Ray, M.U. Chowdhury, J.H. Abawajy, Secure object tracking protocol for the internet of things, IEEE Internet Things J. 3 (4) (2016) 544–553.

[9] Y. Waizumi, M. Omachi, K. Tanaka, On-demand color calibration for pedestrian tracking in nonoverlapping fields of view, IEEE Internet Things J. 4 (2) (2016) 320–329.

[10] J. Lin, W. Yu, X. Yang, P. Zhao, H. Zhang, W. Zhao, An edge computing based public vehicle system for smart transportation, IEEE Trans. Veh. Technol. 69 (11) (2020) 12635–12651.

[11] X. Zheng, Z. Cai, Privacy-preserved data sharing towards multiple parties in industrial iots, IEEE J. Sel. Areas Commun. 38 (5) (2020) 968–979.

[12] X. Liu, C. Qian, W.G. Hatcher, H. Xu, W. Liao, W. Yu, Secure internet of things (iot)-based smart-world critical infrastructures: survey, case study and research opportunities, IEEE Access 7 (2019) 79523–79544.

[13] H. Noguchi, T. Demizu, M. Kataoka, Y. Yamato, Distributed search architecture for object tracking in the internet of things, IEEE Access 6 (2018) 60152–60159.

[14] H. Gu, Z. Ge, E. Cao, M. Chen, T. Wei, X. Fu, S. Hu, A collaborative and sustainable edge-cloud architecture for object tracking with convolutional siamese networks, IEEE Trans. Sustainable Comput. (2019).

[15] J. Li, J. Zhi, W. Hu, L. Wang, A. Yang, Research on the improvement of vision target tracking algorithm for internet of things technology and simple extended application in pellet ore phase, Future Generation Computer Systems (2020).

[16] D. Zhang, Z. He, Y. Qian, J. Wan, D. Li, S. Zhao, Revisiting unknown rfid tag identification in large-scale internet of things, IEEE Wireless Commun. 23 (5) (2016) 24–29.

[17] H. Zhang, Z. Zhang, L. Zhang, Y. Yang, Q. Kang, D. Sun, Object tracking for a smart city using iot and edge computing, Sensors 19 (9) (2019) 1987.

[18] J. Stovall, A. Harris, A. O'Grady, M. Sartipi, Scalable object tracking in smart cities, in: 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019, pp. 3813–3819.

[19] M. Jia, Z. Gao, Q. Guo, Y. Lin, X. Gu, Sparse feature learning for correlation filter tracking toward 5g-enabled tactile internet, IEEE Trans. Ind. Inf. (2019).

[20] D.K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, C. Quek, Video processing from electro-optical sensors for object detection and tracking in a maritime environment: a survey, IEEE Trans. Intell. Transp. Syst. 18 (8) (2017) 1993–2016.

[21] X. Zhu, R. Pan, G. Li, Y. Liu, H. Wang, et al., Network vector autoregression, The Annals of Statistics 45 (3) (2017) 1096–1123.

[22] M. Lanne, J. Luoto, Gmm estimation of non-gaussian structural vector autoregression, Journal of Business & Economic Statistics (2019) 1–13.

[23] F. Ding, F. Wang, L. Xu, M. Wu, Decomposition based least squares iterative identification algorithm for multivariate pseudo-linear arma systems using the data filtering, J Franklin Inst 354 (3) (2017) 1321–1339.

[24] F. Harrou, Y. Sun, M. Madakyaru, B. Bouyedou, An improved multivariate chart using partial least squares with continuous ranked probability score, IEEE Sens J 18 (16) (2018) 6715–6726.

[25] C. Chatfield, H. Xing, The analysis of time series: An introduction with r, CRC press, 2019.

[26] Y. Zheng, Q. Li, Y. Chen, X. Xie, W.-Y. Ma, Understanding mobility based on gps data, in: Proceedings of the 10th international conference on Ubiquitous computing, 2008, pp. 312–321.

[27] Y. Zheng, L. Zhang, X. Xie, W.-Y. Ma, Mining interesting locations and travel sequences from gps trajectories, in: Proceedings of the 18th international conference on World wide web, 2009, pp. 791–800.

[28] Y. Zheng, X. Xie, W.-Y. Ma, et al., Geolife: a collaborative social networking service among user, location and trajectory, IEEE Data Eng. Bull. 33 (2) (2010) 32–39.

[29] J. Lin, W. Yu, X. Yang, Q. Yang, X. Fu, W. Zhao, A novel dynamic en-route decision real-time route guidance scheme in intelligent transportation systems, in: 2015 IEEE 35th International Conference on Distributed Computing Systems, IEEE, 2015, pp. 61–72.

[30] J. Lin, W. Yu, N. Zhang, X. Yang, L. Ge, Data integrity attacks against dynamic route guidance in transportation-based cyber-physical systems: modeling, analysis, and defense, IEEE Trans. Veh. Technol. 67 (9) (2018) 8738–8753.

[31] Y. Wu, Y. Cui, W. Yu, C. Lu, W. Zhao, Modeling and forecasting of timescale network traffic dynamics in m2m communications, in: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), IEEE, 2019, pp. 711–721.

[32] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, K. Huang, Toward an intelligent edge: wireless communication meets machine learning, IEEE Commun. Mag. 58 (1) (2020) 19–25.

[33] B. Zaman, L.M.L. Ramos, D. Romero, B. Beferull-Lozano, Online topology identification from vector autoregressive time series, IEEE Trans. Signal Process. 69 (2020) 210–225.

[34] B. Zaman, L.M.L. Ramos, B. Beferull-Lozano, Dynamic regret analysis for online tracking of time-varying structural equation model topologies, in: 2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA), IEEE, 2020, pp. 939–944.

[35] Y.-H. Han, H.-K. Lim, J.-M. Gil, Hierarchical location caching scheme for mobile object tracking in the internet of things, J. Inf. Process. Syst. 13 (5) (2017) 1410–1429.

[36] T. Wei, X. Zhang, mtrack: High-precision passive tracking using millimeter wave radios, in: Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, 2015, pp. 117–129.

[37] S. Jiang, Z. Lin, Y. Li, Y. Shu, Y. Liu, Flexible high-resolution object detection on edge devices with tunable latency, in: ACM International Conference on Mobile Computing and Networking (MobiCom), 2021.

[38] M.S. Adam, M.H. Anisi, I. Ali, Object tracking sensor networks in smart cities: taxonomy, architecture, applications, research challenges and future directions, Future Generation Computer Systems 107 (2020) 909–923.

[39] S. Kim, S. Kim, S. Kim, D. Han, H.-J. Yoo, A 64.1 mw accurate real-time visual object tracking processor with spatial early stopping on siamese network, IEEE Trans. Circuits Syst. II Express Briefs 68 (5) (2021) 1675–1679.

[40] H. Van Nguyen, H. Rezatofighi, B.-N. Vo, D.C. Ranasinghe, Multi-objective multi-agent planning for jointly discovering and tracking mobile objects, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 7227–7235.

[41] L. Liu, H. Li, M. Gruteser, Edge assisted real-time object detection for mobile augmented reality, in: The 25th Annual International Conference on Mobile Computing and Networking, 2019, pp. 1–16.

[42] T. Zhu, J. Li, Z. Cai, Y. Li, H. Gao, Computation scheduling for wireless powered mobile edge computing networks, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 596–605.

[43] R. Xu, S.Y. Nikouei, Y. Chen, A. Polunchenko, S. Song, C. Deng, T.R. Faughnan, Real-time human objects tracking for smart surveillance at the edge, in: 2018 IEEE International Conference on Communications (ICC), IEEE, 2018, pp. 1–6.

[44] T. Zhu, T. Shi, J. Li, Z. Cai, X. Zhou, Task scheduling in deadline-aware mobile edge computing systems, IEEE Internet Things J. 6 (3) (2018) 4854–4866.

[45] D. Pudasaini, A. Abhari, Scalable object detection, tracking and pattern recognition model using edge computing, in: 2020 Spring Simulation Conference (SpringSim), IEEE, 2020, pp. 1–11.