Technical Disclosure Commons

Defensive Publications Series

December 21, 2018

# MACHINE LEARNING FRAMEWORK FOR PRIORITIZING LOCATION MEASUREMENTS OF MULTIPLE DEVICES

Xu Zhang

Huy Tran

Matt Silverman

Abhishek Mukherji

Vinay Raghuram

***See next page for additional authors***

**Inventor(s)**
Xu Zhang, Huy Tran, Matt Silverman, Abhishek Mukherji, Vinay Raghuram, Abhishek Bhattacharyya, and Santosh Pandey

# MACHINE LEARNING FRAMEWORK FOR PRIORITIZING LOCATION MEASUREMENTS OF MULTIPLE DEVICES

AUTHORS:

Xu Zhang
Huy Tran
Matt Silverman
Abhishek Mukherji
Vinay Raghuram
Abhishek Bhattacharyya
Santosh Pandey

## ABSTRACT

Presented herein is a framework for prioritizing location measurements of multiple client devices. In particular, rather than using a round robin scheduling approach, the techniques presented herein utilize a machine learning block (e.g., random forests) to predict a score for each client device, along with a score-based scheduler.

## DETAILED DESCRIPTION

In conventional indoor location tracking, the locations of clients may be measured in a round robin fashion. The locations of clients are measured one after another until reaching a maximum number of clients that can be measured in one time slot. For example, if each time slot can measure K clients, the location of clients 1 through K are measured in time slot 1, and client K+1 through client 2K are measured in time slot 2. This method leads to inefficient network resource utilization, as different client devices have different behaviors and requirements. For example, some organizations may desire to track the location of certain devices with a higher priority. By reducing the frequency of measurements for stationary clients, mobile clients can be measured more frequently. Additionally, if a device has a low battery level, it can be assigned a high priority in case the device runs out of power. As a further complication, some Internet of Things (IoT) devices only wake up from time to time, complicating the scheduling of location measurements. Present embodiments utilize a machine learning-based framework to predict a priority score for each device, and then use the priority scores to schedule location measurements, thereby addressing the above-identified issues.

Conventional indoor location tracking uses a round robin method to measure the location of different clients. Figure 1 depicts an example of time slots and corresponding assignments of device location measures.

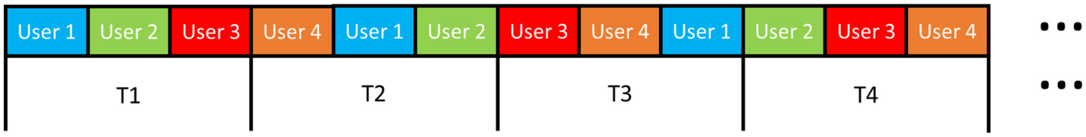

Figure 1

As depicted in Figure 1, there are four clients devices (user 1, user 2, user3, and user 4), but each time slot can only accommodate the measurement of three clients. Thus, user 1, user 2, and user 3 are measured in time slot 1, user 4, user 1, and user 2 in time slot 2, and the like.

In contrast, an optimal way of scheduling location measurement is to give different client devices different weights. Figure 2 depicts another example of time slots and corresponding assignments of device location measures.

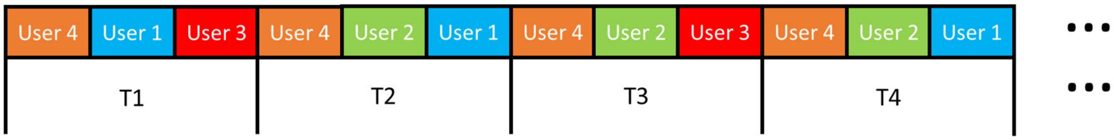

Figure 2

As shown in Figure 2, user 4 may be a valuable device whose location needs to be measured more frequently. Therefore, user 4 is scheduled to be measured at the beginning of each time slot. In some embodiments, each time slot (also referred to as a “dwell”) is set to 250 ms, and can be used to measure at most 18 clients. Therefore, because different clients can have different behaviors and requirements, measuring their location in a round robin manner substantially reduces the network efficiency.

Figure 3 depicts a framework for measuring device locations in accordance with present embodiments.

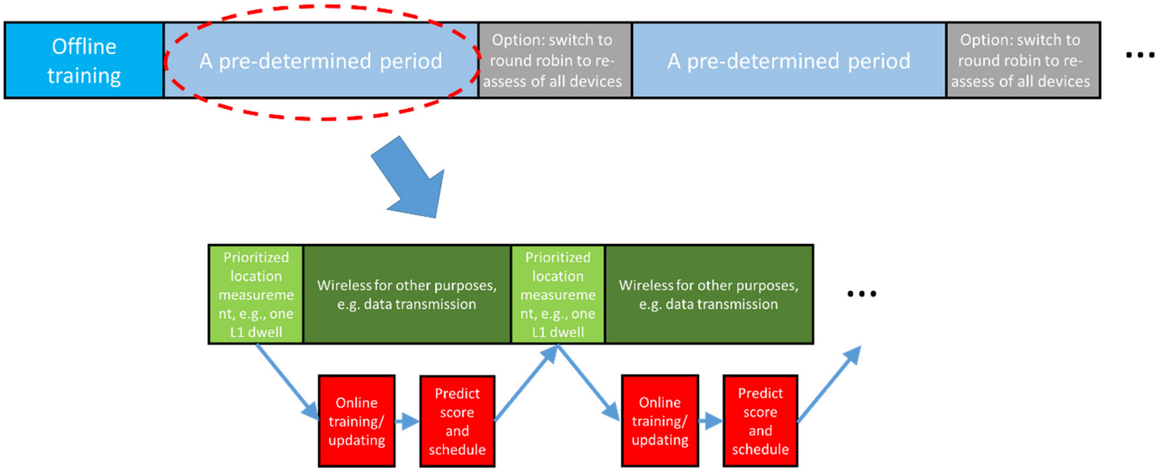
Offline training
A pre-determined period
Option: switch to round robin to re-assess of all devices
A pre-determined period
Option: switch to round robin to re-assess of all devices
...
Prioritized location measurement, e.g., one L1 dwell
Wireless for other purposes, e.g. data transmission
Prioritized location measurement, e.g., one L1 dwell
Wireless for other purposes, e.g. data transmission
...
Online training/ updating
Predict score and schedule
Online training/ updating
Predict score and schedule

Figure 3

A machine learning approach is utilized to compute a score before each dwell (or multiple dwells) for each client whose location needs to be measured. The machine learning algorithm may use a decision tree or random forests. Scores are then fed into a score-based scheduler to compute a schedule for location measurement for the following dwell (or multiple dwells).

Offline training of the machine learning framework that computes the score can be performed initially. Moreover, between each dwell, there will be some amount of time that is consumed by the wireless radios for data transmission. This time may be used to perform online training and refinement. Finally, after certain dwells (either pre-determined or computed on-the-fly), there is an optional period within which the scheduler may return to a round robin approach and measure the location of all clients sequentially. This ensures that all clients' locations are updated after a certain amount of time, and that the machine learning framework can be updated, taking into consideration all current devices in the network.

Figure 4 depicts an example framework including score computation and online training in accordance with present embodiments.

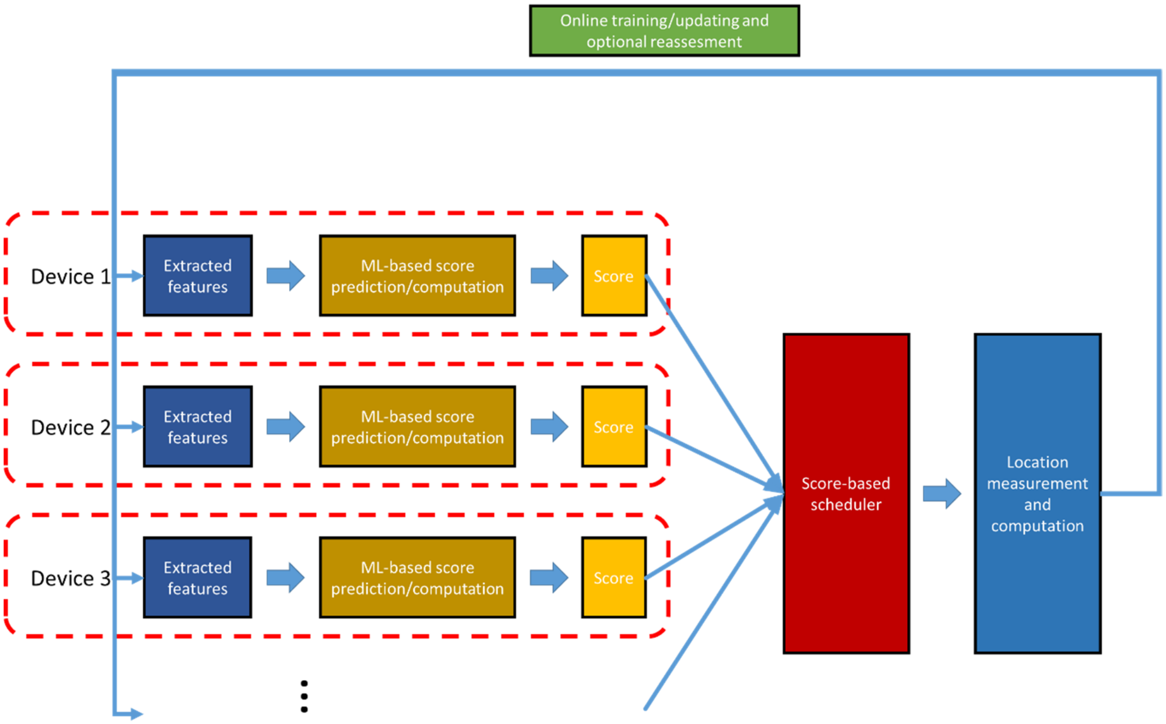
Online training/updating and optional reassesment
Device 1
Extracted features
ML-based score prediction/computation
Score
Device 2
Extracted features
ML-based score prediction/computation
Score
Device 3
Extracted features
ML-based score prediction/computation
Score
Score-based scheduler
Location measurement and computation

Figure 4

In particular, before each dwell, features extracted from each device are fed into the machine learning-based score prediction block, which computes a score for each client. Next, clients are scheduled based on their scores, and locations of some of the clients are measured in the following dwell. Finally, the measurements of these clients are used to update the machine learning framework. In some embodiments, only one machine learning block is employed for every client. In other embodiments, different machine learning blocks may be used for different clients.

Figure 5 depicts inputs and output of a machine learning framework in accordance with present embodiments.

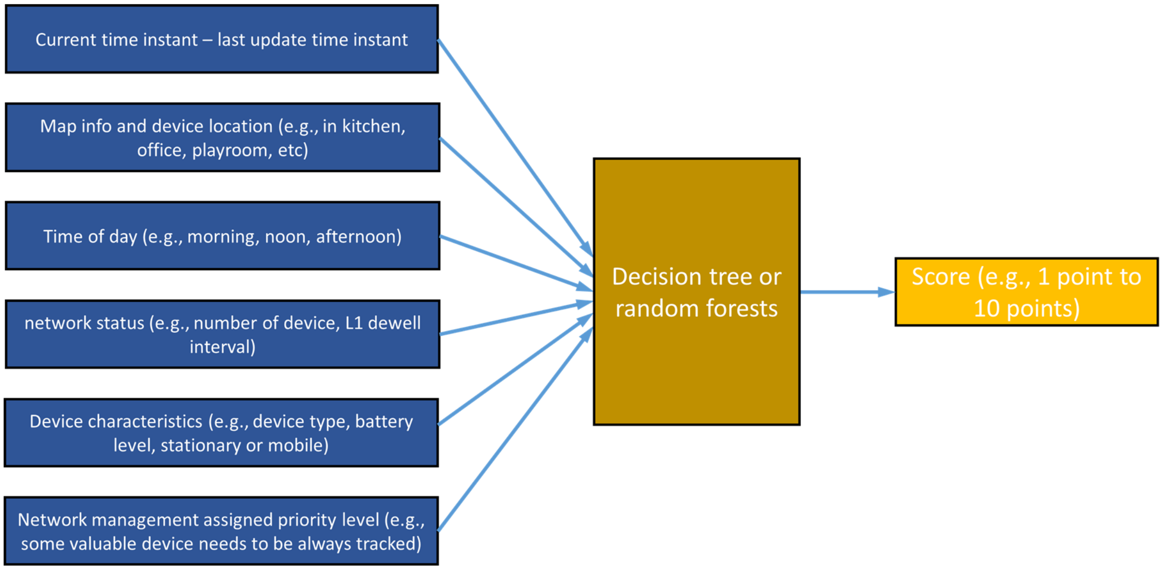
Current time instant – last update time instant
Map info and device location (e.g., in kitchen, office, playroom, etc)
Time of day (e.g., morning, noon, afternoon)
network status (e.g., number of device, L1 dewell interval)
Device characteristics (e.g., device type, battery level, stationary or mobile)
Network management assigned priority level (e.g., some valuable device needs to be always tracked)
Decision tree or random forests
Score (e.g., 1 point to 10 points)

Figure 5

In the depicted example, decision tree or random forests are used in the machine learning block. In other embodiments, other machine learning frameworks, such as artificial neural networks, may be employed. The inputs include descriptions such as the time of data and the device type. Inputs may also include features that may be extracted by the machine learning block. For example, a battery draining pattern may be extracted, as some device's batteries may drain more quickly than others. Extraction of some features may require a recurrent neural network. The output of the machine learning block is a score for the client. In the example, discrete scores ranging from 1 to 10 are used. However, in other embodiments, other features may be used as inputs, and/or there may be a continuous output.

Both offline and online training is included in the machine learning framework. Figure 6 is a block diagram depicting offline and online training.

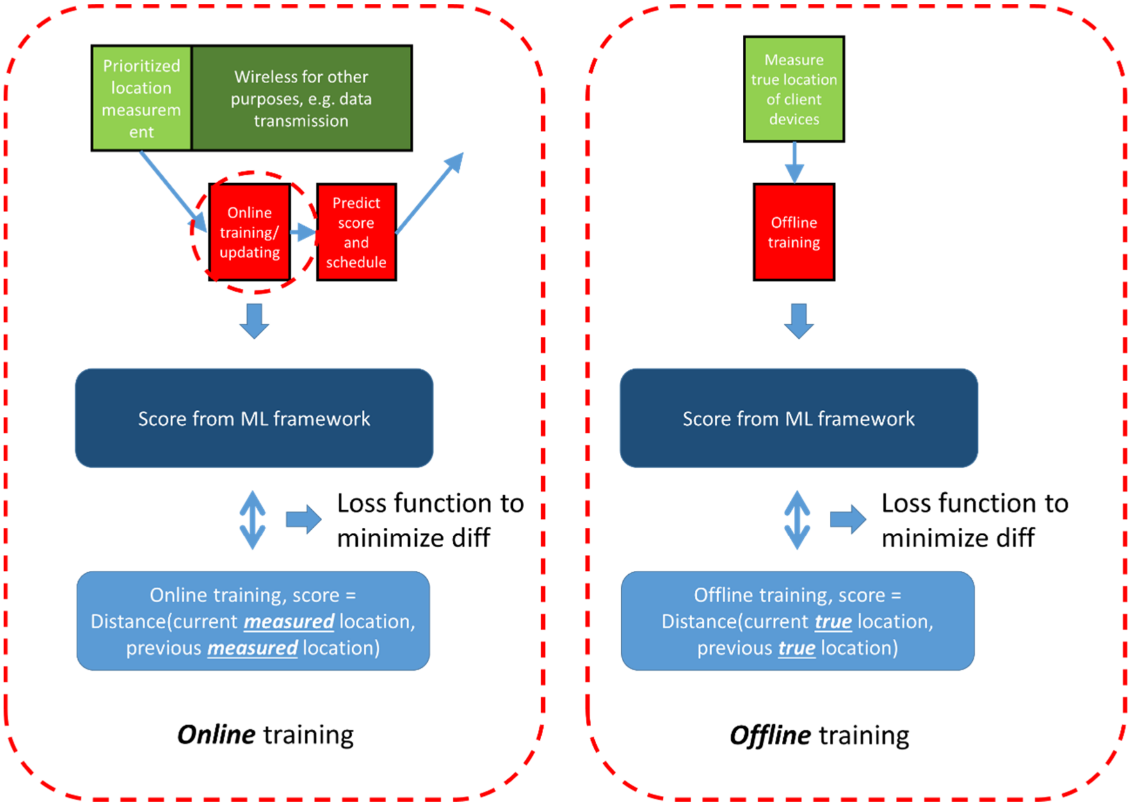
Prioritized location measurem ent
Wireless for other purposes, e.g. data transmission
Online training/ updating
Predict score and schedule
Score from ML framework
Loss function to minimize diff
Online training, score = Distance(current measured location, previous measured location)
Online training
Measure true location of client devices
Offline training
Score from ML framework
Loss function to minimize diff
Offline training, score = Distance(current true location, previous true location)
Offline training

Figure 6

For offline training, a score is computed for each device based on the distance between a current actual location (in current location measurement period) and a previous actual location (in previous location measurement period). For example, scores may be computed using the equation:

computed_score = bound{round[distance(current actual location, previous actual location)]}

In this equation, the round(.) function is used to round the continuous distance to the nearest integer, and the bound(.) function is used to bound the distance between the minimum score and maximum score. Typically, if the distance is small, then an additional measurement is not needed. In comparison, a large distance may indicate that a new measurement is required. In some embodiments, the value (time stamp of current measurement - time stamp of previous measurement) may be omitted in computing the score in the above equation, because it is included as an input to the machine learning framework.

Next, a loss function is calculated based on the computed_score and the output of the machine learning framework. The model is trained to minimize the loss function. The loss may be computed using the equation:

> loss(cross entropy loss) = - label*log(predict) - (1-label)*log(1-predict))

A one vs. all training method may be applied. For each possible score, if the output of the machine learning module matches the score, the predicted value is set to 1, and if not, it is set to 0. In other embodiments in which the score is continuous, the loss may be computed using the equation:

> loss = (computed_score (given the true location is known) - output of the ML framework)$^2$

Online training may be performed similarly to the offline training. After each dwell and measurement of the locations of at least some clients, a score may be computed based on the distance between a current measured location and a previously measured location for each of these clients. A score may be calculated according to the equation:

> score = bound{round[distance(current measured location, previous measured location)]}

While a measured location may contain noises that make the above computed score inaccurate, relative location changes measured (e.g., according to Euclidean distance) by the location system are usually very close to that of the actual locations, since the systematic errors tend to be similar for all location measurements. On the other hand, to reduce noise, location of the same client can be averaged over time, or a Kalman filter may be applied. In general, when location computations are more accurate, the online score prediction is improved.

Compared to offline training, which may include a more diverse training data set (i.e., different device types, different location, different time, etc.), online training data sets are smaller. Furthermore, online training has may be adapted to the specific environment of the current location measurements. Finally, after certain dwells, there is an optional period, within which the scheduler may return to a round robin approach by measuring the location of all clients sequentially. These measurements can be used in the above online training method to update the machine learning framework.