Location Prediction Based Action

Deekshith Allamaneni
Dept. of Electrical & Computer Engineering
Missouri University of Science and Technology, Rolla, Missouri 65409
Email: daqnf@mst.edu

***Abstract*—The aim of this project is to predict the current location of users based on their past location history and take an action depending on the location match. In this project, the GPS sensor in the mobile phones is used to track the current location of the user and the data is logged onto a server at a defined interval. The server processes location coordinates logged over time using a neural network and estimates the current location of the user. The client requests the predicted location from server and takes an action like notifying or alerting the user if the user is not found to be in an incorrent location at that time.**

I. Introduction

Many a times we miss a schedule due to forgetfulness. Reminders and alarms solve this issue to some extent but there can be times when we even forget to set an alarm. This project aims to solve this problem by keeping track of users' location and then generates a prediction of their current location by using neural networks. The current location and the predicted location is compared and necessary action is taken by the client side application.

The main scope of this application is to predict the user's current location. The methods and functions to get an estimation as well as for location comparison are built into the client application and are demostrated. However, the applications of this can be extended with time using the existing client and server infrastructure.

II. Working

A. Client Application

Most smart phones are equipped with a GPS sensor and also provide an API for applications to query its current geographical location. For this project we have designed a mobile application for Android operating system that acquires the current location of the user by using the GPS sensor equipped with the smartphone.

The client has two main functions.

1) Data Logging
2) Location prediction based action

1) Data Logging: The client application acquires the current location of the user via Android API and uses the REST API provided by the server side application to send the location and the time at which the location is acquired to the server.

The Table I shows the data that the clinet saves periodically to the server.

	Latitude	Longitude	Weekday	Hours	Minutes
Range	-90 to +90	-180 to +180	0 to 6	0 to 23	0 to 59

TABLE I
Data stored to server by the client

2) Location prediction based action: The server also provides a HTTP REST API to query the predicted location of the user. The client requests the current predicted location of the user using HTTP GET request. The client also gets the current GPS location from the sensor and takes an action depending on the comparison.

a) Action 1: Report known location but prediction not matched: When the user is at a known location but not at the right time, it notifies the user as shown in Figure 1 and provides an interface for user acknowledgement.

b) Action 2: Unidentified location: When the user is at an unknown location for a certain amount of time, the client side application requests the user to enter the name of this current location. If no name is provided but if the prediction reports that location, it just identifies that location by the coordinates. Figure 2 shows the unidentified location notification and Figure 3 shows the interface to enter a new location.

Notification Interval (sec)	MQTT QoS1	CoAP
5	86	65
30	100	96

TABLE II
Avg Packet Received Ratio

References

[1] Joel L. Fernandes and Ivo C. Lopes, *Performance Evaluation of RESTful Web Services and AMQP Protocol*, Ubiquitous and Future Networks (ICUFN), Da nang. IEEE 2013 Fifth International Conference on, July 2013.

[2] Zanella, A. ; Bui, N. ; Castellani, A. ; Vangelista, L. ; Zorzi, M, Internet of Things for Smart Cities, Internet of Things Journal, IEEE 2014, Volume: 1 , Issue: 1, Page(s): 22 - 32.

[3] Sye Loong Keoh ; Kumar, S.S. ; Tschofenig, H, Securing the Internet of Things: A Standardization Perspective, Internet of Things Journal, IEEE 2014 Volume: 1 , Issue: 3, Page(s): 265 - 275.

Appendix A
CoAP Client

```
import SOAPpy
import time
```

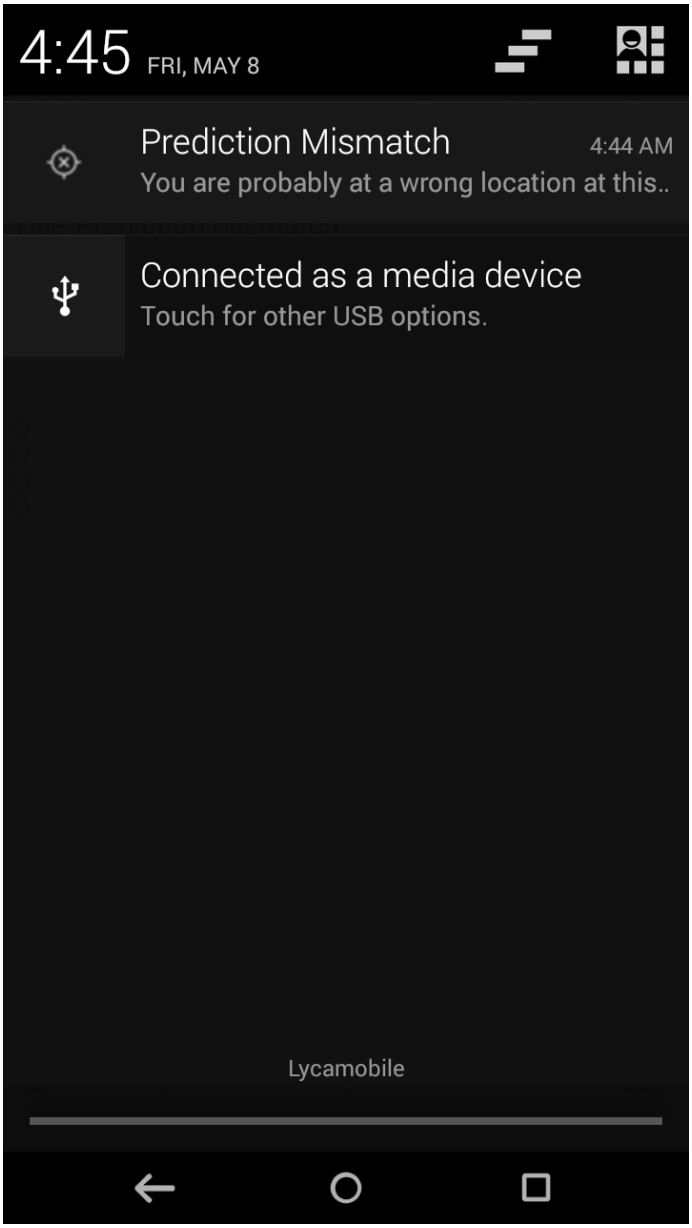

Fig. 1. Screenshot of the client app displaying prediction mismatch notification

Fig. 2. Screenshot of the client app notifying about unknown location

```
i=0
fo = open("coap-results.txt", "w")
while i<1000:
        start_time = time.time()
        server = SOAPpy.SOAPProxy \
        ("http://localhost:8080/")
        time_taken= time.time() \
        - start_time
        fo.write(str("%.10f" % time_taken) \
        +","+str(len(server.hello()))+"\n")
        i+=1
fo.close()
print server.hello()
```

APPENDIX B
COAP SERVER

```
import SOAPpy
def hello():
    return "Hello World!"
server = SOAPpy.SOAPServer \
        (("localhost", 8080))
server.registerFunction(hello)
server.serve_forever()
```

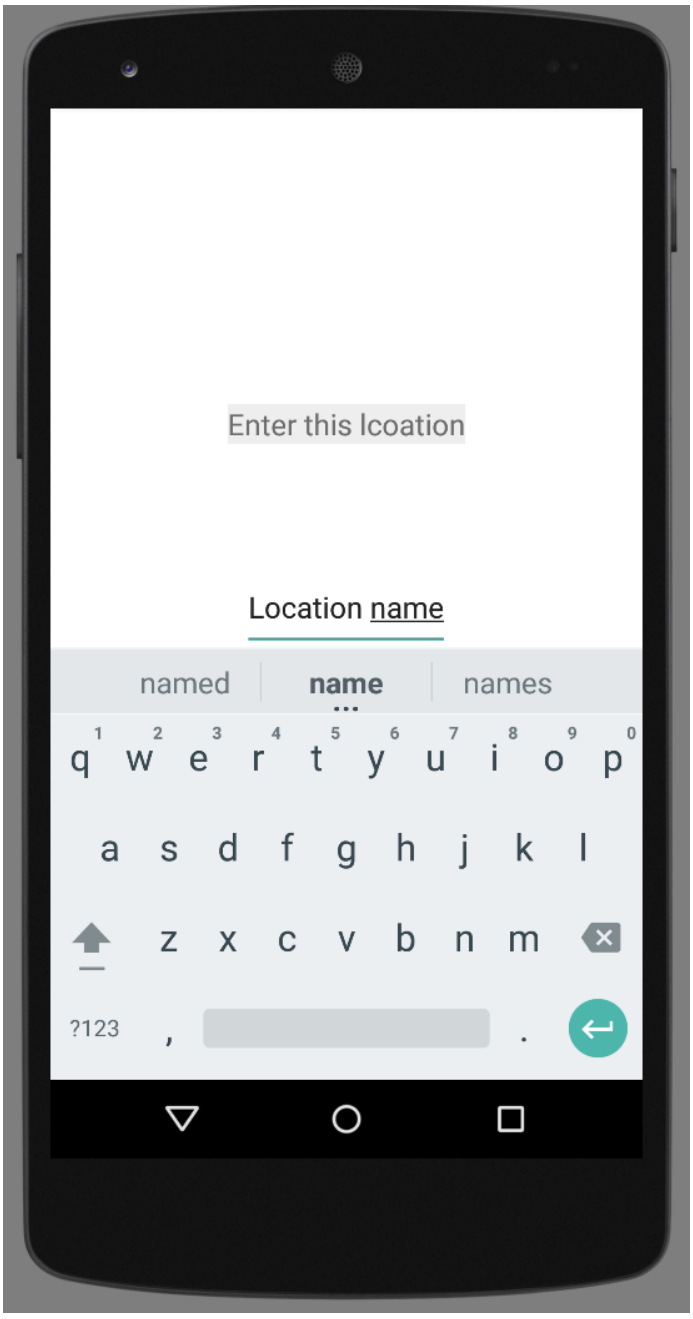

Fig. 3. Screenshot of the client app interface to enter location name