Adaptive WiFi Fingerprinting for Location Approximation

Mohd Fikri Azli bin Abdullah, Khairul Anwar bin Kamarul Hatta, Esther Jeganathan

***Abstract*—**WiFi has become an essential technology that is widely used nowadays. It is famous due to its convenience to be used with mobile devices. This is especially true for Internet users worldwide that use WiFi connections. There are many location based services that are available nowadays which uses Wireless Fidelity (WiFi) signal fingerprinting. A common example that is gaining popularity in this era would be Foursquare. In this work, the WiFi signal would be used to estimate the user or client's location. Similar to GPS, fingerprinting method needs a floor plan to increase the accuracy of location estimation. Still, the factor of inconsistent WiFi signal makes the estimation defer at different time intervals. Given so, an adaptive method is needed to obtain the most accurate signal at all times. WiFi signals are heavily distorted by external factors such as physical objects, radio frequency interference, electrical interference, and environmental factors to name a few. Due to these factors, this work uses a method of reducing the signal noise and estimation using the Nearest Neighbour based on past activities of the signal to increase the signal accuracy up to more than 80%. The repository yet increases the accuracy by using Artificial Neural Network (ANN) pattern matching. The repository acts as the server cum support of the client side application decision. Numerous previous works has adapted the methods of collecting signal strengths in the repository over the years, but mostly were just static. In this work, proposed solutions on how the adaptive method is done to match the signal received to the data in the repository are highlighted. With the said approach, location estimation can be done more accurately. Adaptive update allows the latest location fingerprint to be stored in the repository. Furthermore, any redundant location fingerprints are removed and only the updated version of the fingerprint is stored in the repository. How the location estimation of the user can be predicted would be highlighted more in the proposed solution section. After some studies on previous works, it is found that the Artificial Neural Network is the most feasible method to deploy in updating the repository and making it adaptive. The Artificial Neural Network functions are to do the pattern matching of the WiFi signal to the existing data available in the repository.

***Keywords*—**Adaptive Repository, Artificial Neural Network, Location Estimation, Nearest Neighbour Euclidean Distance, WiFi RSSI Fingerprinting.

I. INTRODUCTION

TODAY, people are living in a fast paced technology-driven era, where the community are seeking for more accuracy in terms of information. There is a pressing demand for location based services such as GPS, Friend Finder, and Foursquare and so on. All these services require a return of the most accurate location. This is made possible by the WiFi fingerprinting. Apart from WiFi technology, there are many options on implementing location approximation technologies. Global Positioning System (GPS) is the main leading technology in this area with many popular applications such as Garmin, this is because its consistency and reliability of information provided. Another close competitor is RFID which is much cheaper in terms of cost of equipment. Meanwhile, Bluetooth is famous for transferring data from mobile to mobile using ad hoc. However, in this work, WiFi technology is used. There is a major drawback in obtaining location based on WiFi signal strengths. This is because, signals are faced with major challenges such as reflection, noise, attenuation, and weather which will cause signals to fluctuate at different time intervals. Crowded places also interfere with signal transmission causing the stability of the signal to waver. A more consistent signal is needed and a new technique for fingerprinting is demanded to increase the accuracy. The implementation of WiFi Fingerprinting is rarely found in the existing application. The problem with implementing it is the precision of the location approximation is doubted as the implementation is still in its early stage. There are other factors which also contributed to the deployment of WiFi Fingerprinting, and this could be due to signals which are prone to deviation such as weather fluctuations and other factors.

Despite the disadvantages posed by WiFi technology, it is the easiest to implement as most mobile applications such as smartphones and laptop have access to the technology. Connecting to the WiFi is relatively simple too. Hence, WiFi Fingerprinting is used in this work for location approximation. Having the cons of WiFi technology in mind, the proposed solution for this work is that the mobile application captures the signal and smoothest the signal using the smoothing algorithm to get the average rate of signal and then send to the repository for pattern matching. The smoothing process would reduce and minimize noise and outliers who may affect signal accuracy. Location fingerprints which are collected would be stored in a repository. As location fingerprints may defer from time to time due to the nature of WiFi signals, an adaptive repository is needed to be constructed. For ultimate accuracy, the return location estimation from the repository would be compared to the floor plan using Nearest Neighbour estimation in the mobile application.

Mohd Fikri Azli bin Abdullah is with the Faculty of Information Science & Technology, Multimedia University, 75450 Melaka, Malaysia (phone: 06-2523381; fax: 06-2318840; e-mail: mfikriazli.abdullah@mmu.edu.my).

Khairul Anwar bin Kamarul Hatta and Esther Jeganathan are with the Faculty of Information Science & Technology, Multimedia University, 75450 Melaka, Malaysia (e-mail: khairulanwar09@gmail.com, estherjeganathan@gmail.com).

Another proposed solution for this work is a method of Artificial Neural Network (ANN) to make the repository adaptive. ANN was chosen as it is relatively easy to implement. This method matches the signal which is obtained from the client side with the data in the repository. A set of user input is trained with the ANN to create a desired output, which is the location. ANN basically receives the input then processes the signal inputs in the hidden layer then an output is obtained. ANN inputs are based on the weights of the signal. A higher weight would result in a stronger input. By adjusting these weights, an output can be returned for specific inputs. The process of adjusting these weights is known as learning or training. When the pattern is matched, the server will return the location approximation to the client. ANN models have proven to be useful and efficient in generating logical outputs based on trained inputs such as satellite image processing, identifying myocardial infarction in patients and so on. However, as ANN involves training of recognizing sets of inputs, this may be time consuming and incurs high cost while developing training sets. Hence to reduce the intensity of the problem, the proposed solution of the ANN architecture consists of three input nodes, five hidden nodes and four output nodes. The ANN approach is very appropriate as any changes in the fingerprint would be done automatically. The repository would be updated automatically whenever the fingerprint that is stored in the database does not match. Existing fingerprints in the repository are updated to the very latest pattern match with the location. Thus, there are no redundant location fingerprint data in the repository. Hence, location determination can be done more accurately.

II. Previous Work

A. RSSI WiFi Location Fingerprinting

Wireless technology has benefited many users today by providing a range of services available on the Internet. Smartphone users are especially taking advantage of the wireless capability of the phone to gain access to the Internet. Now, users can have access to the latest information, increase communication efficiency, and thus, create a closer bond between humans like never before. As technology advances from time to time, people are clamouring for more accuracy and efficiency in services, namely location based services. Most of the time, users are prompt to provide a feedback to the server for updating purposes and this can be annoying in the long run. With that in mind, this work focuses on automatic updates that makes location based services to be hassle free in addition to accurate location information anytime and anywhere.

Some background study on previous works has been highlighted in the following paragraphs as their implementation of location estimation contributed to the overall idea of this work.

Dik Lun Lee and Qixua Chen [1] stated in 2007 that the mobility pattern of the user is the focus instead of the signal strength factor. Thus, WHAM!, a mobile-based WiFi Localization method was implemented in three main implementation plan. The first implementation plan consists of knowledge of the floor model of the implementation area. The second step is to continuously track the user location and lastly, back-tracking from the current location to the previous location to resolve location ambiguities. This method continuously records a signal strength values from each access point at the user's location with fixed time interval. The smoothing method done by WHAM! is adapted in this work, as this method is known for reducing noise and outliers of WiFi signal. For the partitioning of data, since the signals collected are long, the data is partitioned before it can be mapped into zones on the floor plan. The purpose of this partitioning is to separate the collected data to several segments.

Streamspin architecture server [2] holds a repository of radio maps of indoor localization enabled buildings. Each building is identified by MAC address of access point. Radio map is downloaded to user's device on an automatic basis. Streamspin distributed the task of building radio maps to its users. The users indicate their position by clicking on a graphical building map. The resulting fingerprint is uploaded to the radio map repository where it is added to the fingerprint vector for the chosen location. While WiFi triangulation's [3] goal is to map RSSI as a function of distance. This method requires a steep linear characterization curve in order to be properly implemented. Functions describing these curves are then used with live RSSI values as input to generate an (x,y) location prediction. This method was considered first due to its relatively simple implementation.

Wendong Xiao, Wei Ni and Yue Khing Toh [4] highlighted the use of WiFi Fingerprinting along with Inertial Sensing to increase the accuracy of the location approximation in indoor positioning in the year 2004. A new technique for WiFi Fingerprinting was highlighted in their work. The technique used was comparing the WiFi tag with a region-based group as a reference instead of using an individual reference point. The integration of inertial sensing is for determining the orientation and the acceleration of the moving mobile platform. This is then integrated with the WiFi Fingerprinting algorithm that is in this previous work, Kalman Filter which is believed to estimate the instantaneous state of a linear dynamic system. This work conclude in declaring that the use of region-based fingerprinting and Kalman Filter along with the integration of inertial sensing has increase the accuracy of the positioning result. Wendong Xiao also stated that the future work would be testing with a multipath signal environment that is one of the main factors affecting the accuracy of the WiFi Fingerprinting location approximation.

Justin Stook [5] described that the need of a geometric map is not needed in the WiFi Fingerprinting. From studying previous work, Kalman Filter and Weighted Nearest Neighbour provided a more accurate positioning result for indoor localization. The technique used for the WiFi Fingerprinting in this work implements two methods, such as the count method with search space and least sum of square method for location determination. The first method uses the concept of counting the number of fingerprint matches until a search space is discovered, meanwhile, the second method is done by calculating the difference between the actual fingerprint and the surveyed fingerprint. Still both of the method poses some disadvantages, that is, it requires all identifiable nodes to have an equal amount of Acess Points (AP) and the sensitivity of WiFi Fingerprinting when receiving a bad reception of signal. So this work comes with a new solution to solve the method weaknesses that is adding a prediction of the future location which can support the location determination of the two methods. This is then shown by Fig. 1.

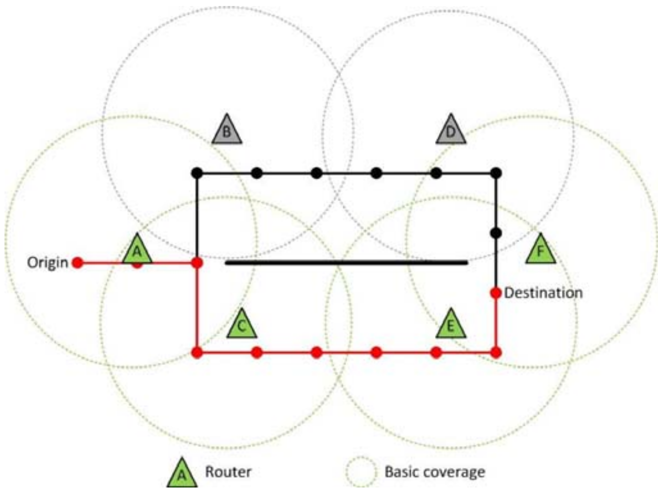


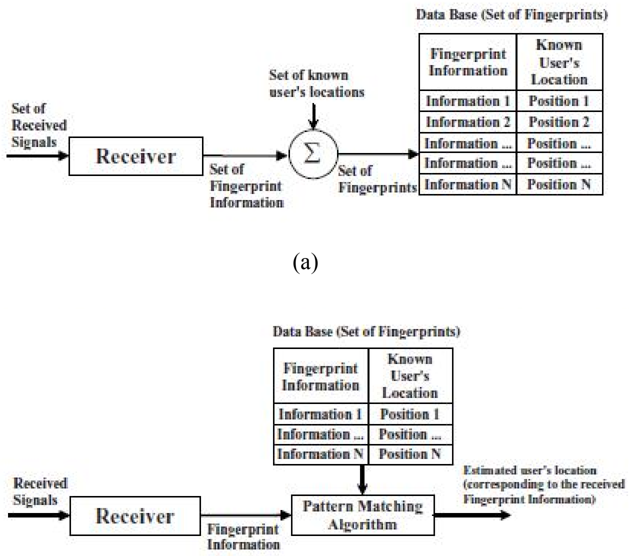
Fig. 1 Prediction Method [5]

The project is then move to the next phase that is, navigation algorithm need to be determined. According to this previous work, Djikstra algorithm was used to calculate the shortest path from the existing fingerprint node to create a direction set for the user to navigate using the application. However, there are limitations, and some of its limitation is it required a lot of pre-processed work and needs to be constantly updated to name a few.

B. Adaptive Repository of WiFi Signals

Location fingerprints may vary from time to time due to the WiFi signal's nature. Thus, an adaptive repository is needed to store the updated location fingerprints to ensure accuracy in location determination of the user. Repositories are conventionally static and updates are done manually. This results in redundant data in the repository. To solve the redundancy, many methods are used in previous work to make the repository adaptive. Previous work has been conducted to uncover the methods used. Following paragraphs highlights the previous work done to make the repository adaptive.

Milos N. Borenovic and Aleksandar M. Neskovic [6] mentioned that ANN is a mechanism used for pattern matching inside the repository. This is similar to the proposed solution in this work. ANN works adaptively based on the condition assigned to it. In location determination in a WLAN environment by use of artificial neural networks and space partitioning work, it is stated that ANN is useful in space partitioning whereas it can be divided to several nodes to increase efficiency in pattern matching for the radio map provided. According to Milos N Borenovic and Aleksandar M Neskovic, [6] ANN is also an efficient way for matching patterns as variation of inputs can be trained to gain the best output. The efficiency is proven in many cases as the ANN holds the power of random variable input pattern matching that enhance the possibilities of higher chances to estimate the location for this work based on variation of signal. Pattern matching involved many requirements as well as expected results to be near as possible to the intended result. Another detail description regarding pattern matching ANN is shows in Fig. 2.



(a)

(b)

Fig. 2 Process of Geolocation [6]

Fig. 2 shows the process of geolocation using received signal's fingerprint. The (a) refers to the off-line phase and the (b) refers to the real time phase [6]. The typical architecture of ANN is depicted in the Fig. 3.

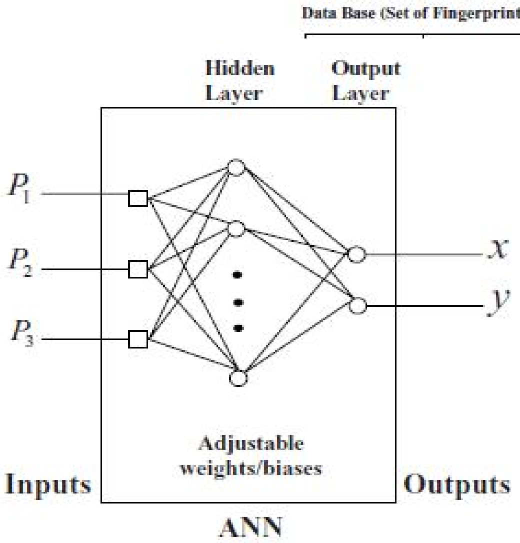


Fig. 3 Architecture of ANN [6]

Benjamin W. Charrow [7] describes the Organic Indoor Localization. The system consists of both server and client but is separated as a unique component. The server in the system is a repository of fingerprints made by clients. When location cannot be determined, user is required to update location and the updates are sent to server. The repository implemented is a stand-alone server. The repository stores, scans and binds location fingerprints. This approach is different from the proposed solution of this work as user updates are not implemented.

Philipp Bolliger in his work, Redpin [8] also requires user involvement, similar to Benjamin W. Charrow's [7] work. However, more user involvement is needed in the Redpin system as users are also required to train while using the system. The Redpin system is adaptive to changes of signals due to the environment such as weather or access points that are replaced. Since the system is adaptive, the users are to manage and create locations. The architecture of Redpin is basically divided into two; the Sniffer component that gathers WiFi signals and creates the fingerprint; the Locator component that stores measured fingerprints in a repository. The Locator contains algorithm to locate a mobile device. The uniqueness of the Locator lies on its versatility. The Locator component is not restricted on a central server as it can be functional on a mobile device as a separate entity. The server part of Redpin provides a service to store fingerprints in a central database. This concept is similar to this work. Fig. 4 illustrates the architecture of Redpin system.

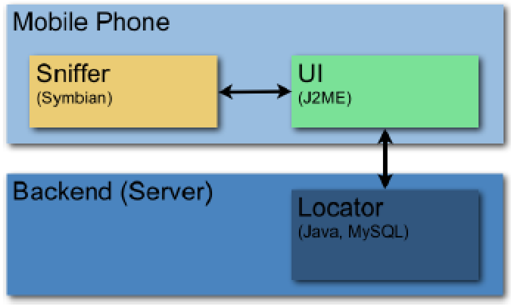


Fig. 4 Redpin system architecture overview [8]

Anthony LaMarcal et al., [9] contribute their idea to the Place Lab system. Place Lab is a location estimation system which collects radio transmissions emitted by access points, mobile devices and GSM cell towers. The Place Lab architecture are comprised of radio beacons in the environment, databases that hold information beacon's locations, and the place lab clients that uses information to estimate current location. Place Lab devices need to interact with radio beacons to learn their IDs. Place Lab clients do not need to transmit data to determine location. Place Lab has a beacon databases that serves the beacon location information to client devices. If Place Lab does not know about a beacon, there is no location estimation. Multiple beacon databases are enabled and privacy is not specified. Fig. 5 is the key components that make up the Place Lab architecture.

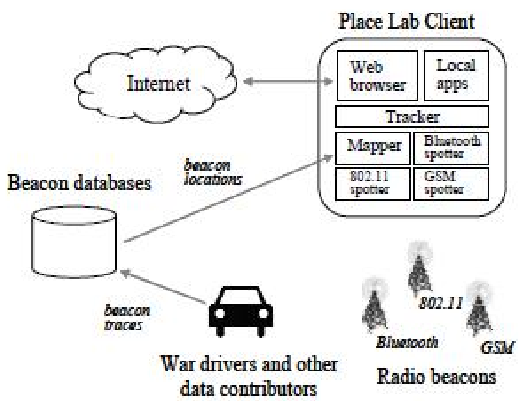


Fig. 5 Key components in the Place Lab Architecture [9]

Place Lab differs from the proposed solution of this work as the signal used to determine the location of the user is GPS, GSM and 802.11.

The Ekahau Positioning Engine Server [10] is a Java-based server that is similar to this project. The Ekahau Positioning Engine calculates and returns the location of WiFi devices via the Ekahau API. Fig. 6 shows how Ekahau works.

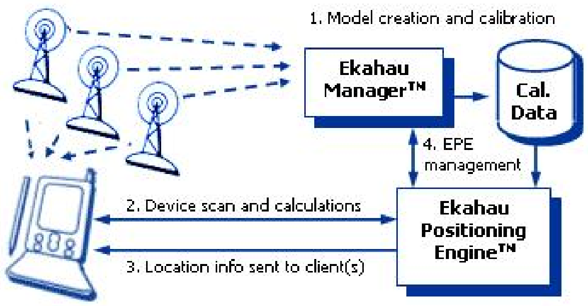


Fig. 6 How Ekahau Works [10]

Ekahau Positioning Engine (EPE) allows PC, PDA and client applications have tracking features, thus having location estimation. The EPE works in any wireless local area network and locates over 100 devices per second. EPE provides a software based system that enables location based applications to return location to user. The user device that system tracks must be running the client software. EPE calculates the user device's locations and the Ekahau Manager is a platform for creating positioning models, tracking devices, and analyzes positioning accuracy.

III. Proposed Solution

From much study of previous works, a better solution was derived weighing the limitations and advantages found in the previous work. WiFi Fingerprinting needed to be implemented with an adaptive method to ensure the accuracy of result. The solution to implement the adaptive method is by using Artificial Neural Network (ANN) to update the repository. Since the objective of this work is to provide location based services with minimum hassle and accuracy of result uncompromised, user feedback for updating the repository is omitted. Following are the proposed solution that is divided into the client and the server part.

A. Client Mobile

The proposed solution for the client is by improving the WiFi signal that is captured. By nature, WiFi signals are prone to distortion due to noise and outliers. Adapted by WHAM! [1], this work would adopt the smoothing process which would remove noise and outliers found in the WiFi signal. The smoothing process would help to increase the accuracy and consistency of the signal even when there are constraints such weather, population and other related factors that can cause noise and distorted signals. The smoothing process is considered to be the best method pertaining to signal accuracy. The flowchart in Fig. 7 summarizes the Client Mobile process.

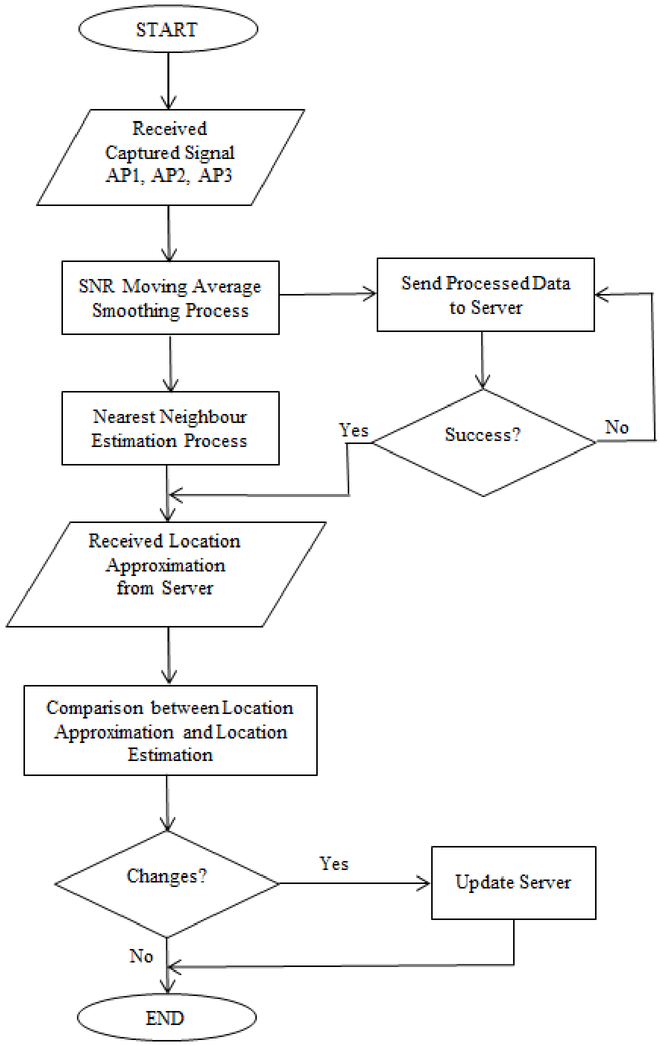


Fig. 7 Client Process Flowchart

Fig. 7 shows that three signals would be captured for processing whereby those three signals is taken based on the strongest signal on the current fingerprint location. If the signal captured can't satisfy the requirement, for example, only two signals are detected in the current area, another signal with a null value would be inserted to satisfy the requirement. In the scenario where there are no signal detected at all, the application would notify the user that there are no signal available and would prompt the user for a rescan or to relocate from the current location. Another important method that is implemented in the signal capturing is that for every signal, three period of two second signal is taken and then later would be processed for smoothing purpose. The application will process, and take the average of the signal for those three instances of periodic signal captured. The following algorithms are used: i) Signal to noise ratio (SNR), Moving Average Algorithm [11] is used to get a more constant rate for the captured signal by reducing the noise and taking the average of the signal. Following shows the equation to calculate SNR algorithm (1):

$$(Y_k)_s = \sum\nolimits_{i=-n}^{i=n} Y_{k+i} / (2n+1) \quad (1)$$

$(Y_k)_s$ stands for smoothed point whereas $\sum\nolimits_{i=-n}^{i=n} Y_{k+i}$ will be the result of the smoothing process, n is number of periodic signal per Access Point and $(2n+1)$ would be the filter width use for measurement of every periodic signal per Access Point average point.

The algorithm will use the value of those three different periodic signal captured as the data needed to calculate the average of each signal to enhance the consistency of the signal variation. It will then be sent through parallel process, which will be sent to the server to get the approximation of the location that is held in the repository based on the smooth signal matching and another process would be the estimation process that used ii) Nearest Neighbour (Euclidean Distance) [3] is one of the most precise algorithms used for location estimation. The algorithm estimates the next node or the nearest node based on a real time experiment.

Nearest Neighbour calculation consists of calculating the nearest possible location from the signal captured. The result of calculation will then be compared with the radio map embedded in the client application to get a better precise estimation. This result will then be compared along with another result that is the approximation result obtained from the repository. In the comparison process, a set of tolerance for signal variation result will be set. The parameter is used to determine the accuracy of the result. The purpose of estimation is to apply the adaptive method where it detects the current location using a simplified radio map that is embedded inside the client application. The radio map is not as accurate as the server where it is just consists of fingerprint node without any fixed value for every node. The server in other cases has a better pattern matching but lack in the sense where it contains a static data that may not be accurate. Combining these two methods, the adaptive method is achieved where the radio map estimation is used to update the server based on the static data of the server. If the estimation value is off its limit, the server will reject the changes and use the current information whereas if the estimation is compatible with the static value in the server, the server will then update the old data to the current one. Compatible ranges are where the value of signal transmissions is logic. The result would be displayed to the user of the mobile application. For a better understanding, Fig. 8 illustrates the overall process.

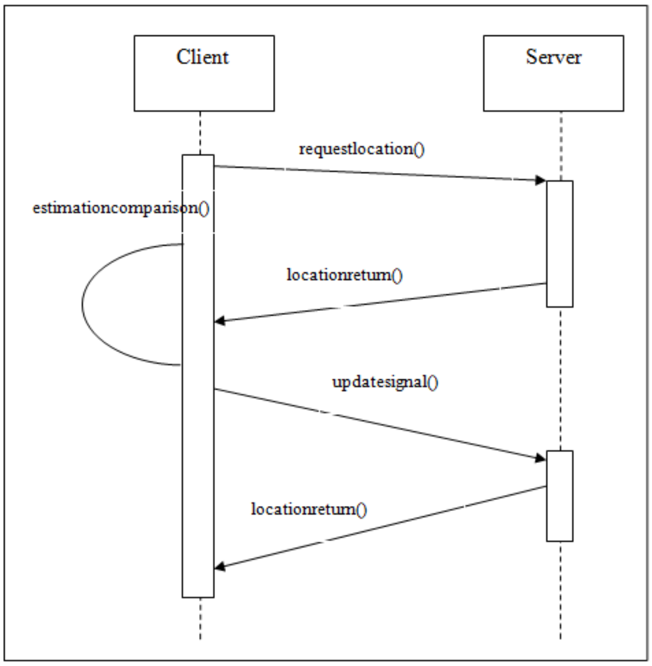


Fig. 8 Timing Diagram Client Process

Fig. 8 illustrates the timing of the process for the communication between the client application and the server. The client would be the first to invoke a process called *requestlocation()* to request the location from the server based on the signal information sent along the process call. The server will then process the matching and invoke a *locationreturn()* process to reply client request. While waiting for the location information from the server, the client had execute a process that run within the client system which is the *estimationcomparison()* that estimate the next location based on the signal input. When the location matching from the server is retrieved, the *estimationcomparison()* process will compare the result with the estimation and invoke a *updatesignal()* process that will update the server information. The server will then return the new location to the client. The location will be displayed without updating process if the result of comparison is the same.

B. Adaptive Server

Repository or known as servers functions as storing the WiFi fingerprints. The main processing of location is actually made in the server whereby the server holds the most recent and updated location fingerprint. Location approximation is done based on pattern matching of the data from the processed data from the client application and the existing data that is stored inside the server. The server uses the Artificial Neural Network architecture to produce the output. The ANN approach enhances the pattern matching based on trained inputs to produce the location of the user. The server would be updated automatically whenever there are changes detected by the client application. The changes are detected when the pattern defers from the client application.

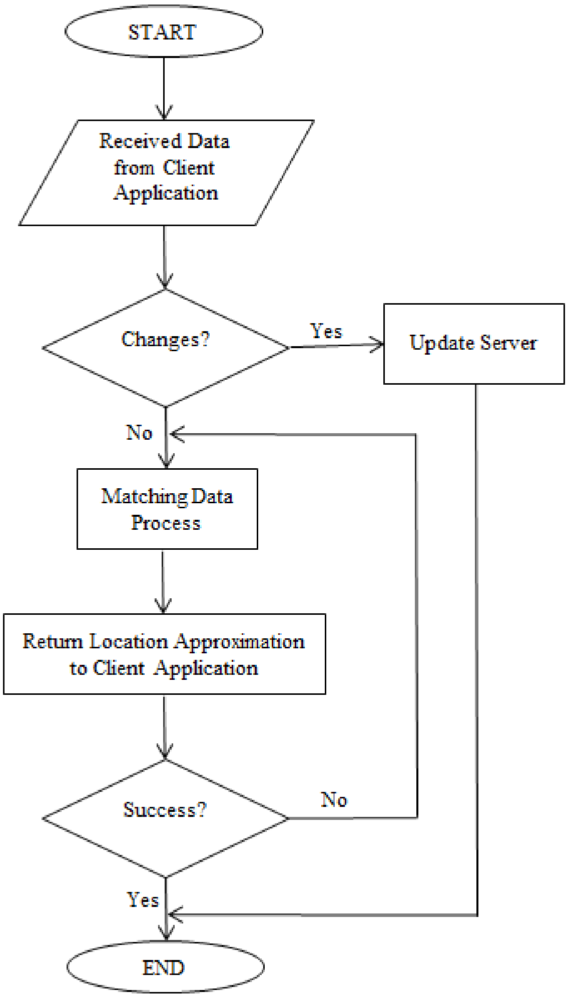


Fig. 9 Server Process Flowchart

Fig. 9 shows the process flow of the server. The process starts by receiving process signal from the client. It is in the form of data which contains either a request to attain location approximation or the request to update the server due to changes in signal is detected for any fingerprint nodes. If there is a change, the server will automatically update and refresh the current data to the updated version for the purpose of real time implementation. This is the adaptive part of the server which requires updates as soon as changes are detected from the client application. The data would be stored in the server. As ANN is an automatic update, this fulfills the objective of the work, which is to provide hassle free location based services whereby no user involvement is needed to update the server. Clients are able to query location at leisure without being bugged at prompts for updating location.

When there is a request in the form of attaining the location approximation, the server would continue with the matching data processes. This is where the server will match the pattern of the data to attain the location for the current fingerprint node. The perks of ANN are the tolerance of signal variation as the ANN is trained within an acceptable range of input. For example, if signal strength is recorded at -98db is stored in the repository for a current location, the ANN would be trained to tolerate up to -88db for the maximum and -108db of signal variation for the current fingerprint node. The signal strength will then determine the location and return the result to the client application. If the retransmit result fails to reach the client, the server will resend the location approximation until the client application receives the result.

This work proposes the ANN to match the pattern of the WiFi fingerprint obtained from the client. In the ANN architecture, a connection between the WiFi fingerprint and location must first be established. The ANN serves as a classification, control and function approximation for the signal.

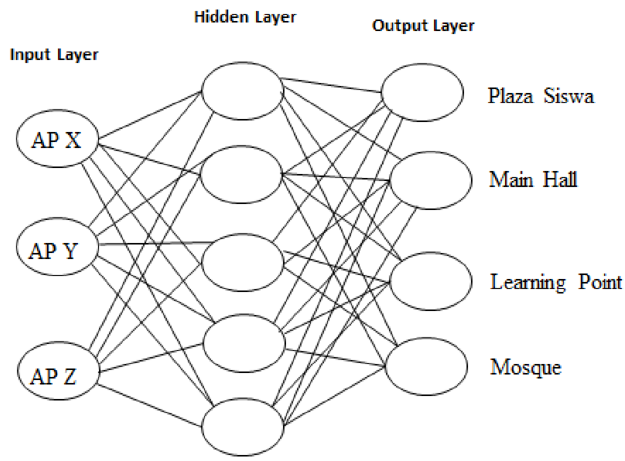


Fig. 10 Proposed Artificial Neural Network Architecture

Fig. 10 shows the proposed artificial neural network architecture. To train a network can be time consuming and costly, thus, the proposed architecture only consists of three input nodes, five hidden nodes and four output nodes. Three inputs with variable APX, APY and APZ are based on the received data signal captured from the access point. The Multi-Layer Perceptron (MLP) consists of five nodes. The ANN uses a Feed forward Algorithm which directs signal in a forward manner. The signal from the input layer reaches the hidden layer, MLP, which would process the inputs and forwards to the output nodes. The output layer contains the location of the fingerprint nodes that is trained. The ANN is trained based on the data tabulated in Table I. The ANN is believed to be the solution for making the repository adaptive and reliable in terms of location estimation. ANN can tolerate variation of signals due to the training of range of signal inputs.

Table I tabulates the values of signal strength of WiFi when a user queries for location. The node receives WiFi signals from AP1, AP2, AP3, AP4 and AP5 where AP refers to Access Point.

TABLE I
PROPOSED TABLE OF LOCATION VS SIGNAL STRENGTH

Location /Signal	Excellent		Medium		Weak	
	Node	Value (DB)	Node	Value (DB)	Node	Value (DB)
Plaza Siswa	AP1	-98	AP2	-95	AP3	-90
Main Hall	AP2	-85	AP1	-83	AP3	-80
Learning Point	AP2	-77	AP3	-75	AP1	-74
Mosque	AP5	-68	AP4	-65	AP3	-60

In this scenario, a user is querying location using a mobile application. The WiFi signal that is strongest is determined as the user's current location, which is Plaza Siswa. The WiFi signal strength is divided to excellent, medium and weak. A stronger feedback of signal value in decibel is returned to indicate the location of the user and tabulated in the table as shown. For example, -98db detected by Access Point 1 identifies the location of the user which is at Plaza Siswa. Being further away from the location would result in a lower value of signal which would be deemed as the most unlikely location of the user, for example in this scenario, the location Mosque, which returns a weak WiFi signal value of -60.

IV. Conclusion and Future Work

In conclusion, the proposed solution or method to adapt to WiFi signal changes is to use Artificial Neural Network (ANN) pattern matching comparison with the Nearest Neighbour estimation to check for changes everytime user request for location approximation. The method allows user to obtain the most accurate location possible. With the proposed solution, the conventional static repository is replaced with the adaptive updating method.

The algorithm for processing the average signal is to make the signal variance become more consistent. The future work would be to find a more suitable algorithm to enhance the WiFi signal capturing and pattern matching so that the accuracy is still the same or even better although there are signal changes. In the future, validation for signal matching may be included to make the accuracy of location determination at its optimum performance. With the validation process, signals which are truly valid only are allowed to be updated in the repository.

References

[1] Lee, Dik Lun, and Qiuxia Chen, "A model-based Wifi localization method," Proceedings of the 2nd international conference on Scalable information systems ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007.
[2] Jensen, Christian S., Carmen Ruiz Vicente, and Rico Wind, "User-generated content: The case for mobile services," IEEE Computer, vol. 41, pp. 116-118, 2008.
[3] Michael Quan, Eduardo Navarro, and Benjamin Peuker, "Wi-Fi Localization Using RSSI Fingerprinting," California Polytechnic State University 1 Grand Avenue, San Luis Obispo, CA, 2010.
[4] Xiao, Wendong, Wei Ni, and Yue Khing Toh, "Integrated Wi-Fi fingerprinting and inertial sensing for indoor positioning," International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2011.
[5] Stook, Justin, "Planning an indoor navigation service for a smartphone with Wi-Fi fingerprinting localization," Diss. Master Thesis, TU Delft, The Netherlands, 2011.
[6] Borenović, Miloš N., and Aleksandar M. Nešković, "Positioning in WLAN environment by use of artificial neural networks and space partitioning," Annals of Telecommunications, vol. 64, pp. 665-676, 2009.
[7] Charrow, Benjamin W., "Organic indoor location: infrastructure and applications," Diss. Massachusetts Institute of Technology, 2010.
[8] Bolliger, Philipp., "Redpin - adaptive, zero-configuration indoor localization through user collaboration," Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments, pp. 55-60, 2008.
[9] LaMarca, Anthony, et al., "Place lab: Device positioning using radio beacons in the wild," Pervasive Computing, pp. 116-133, 2005.
[10] Ekahau Positioning Engine, 2003 (available online at https://whereabouts.eecs.umich.edu).
[11] Constantinos E. Efstathiou, "Signal Smoothing Algorithms," (available online at http://www.chem.uoa.gr/applets/appletsmooth/appl_smooth2.html).