An Experimental Study on the Effect of Pattern Recognition Parameters on the Accuracy of Wireless-Based Task Time Estimation

Andrew W. Muyanja, Tanawat Atichat, J. David Porter[1]

School of Mechanical, Industrial and Manufacturing Engineering, Oregon State University, Corvallis, OR 97331 USA

Task time estimation is a core industrial engineering discipline. However, the process to collect the required data is manually intensive and tedious, thus making it expensive to keep the data current. Radio frequency signals have been used to automate the required data collection in some applications. However, such radio frequency data is subject to systemic and random noise, leading to a reduction in the accuracy of the task time estimation. This research investigates the use of a pattern recognition method, the *k*-nearest-neighbor algorithm, to improve the accuracy of task time estimation in a simulated assembly work area. The results indicate that the parameters of the kNN algorithm can be experimentally tuned to improve the accuracy and to dramatically reduce the necessary computational time and the costs of performing real-time task time estimation.

Keywords: Data mining; *k*-nearest-neighbor; pattern recognition; radio frequency; task time estimation; wireless sensor networks; work measurement.

Subject classification code: Operations Engineering

1. Introduction

Task time estimation is important in manufacturing as it is one of the core elements of industrial engineering tasks such as workstation layout, capacity planning, cost estimation and line balancing. However, traditional task time estimation techniques such as work sampling and time studies are time consuming and tedious.

A novel approach to task time estimation involves monitoring the strength of a radio frequency (RF) signal within a wireless sensor network (WSN) to estimate an operator's position within a workstation in real-time and then use this position information to derive task

[1] Corresponding author: david.porter@oregonstate.edu

duration (Atichat, 2011). A key part of this approach is to utilize a pattern recognition technique called *location fingerprinting* based on the *k-nearest-neighbor* (kNN) classification algorithm to analyze and classify the data generated by the WSN, and then use that processed data to estimate the task time at a specific location. The location fingerprinting process consists of two phases: the offline data collection phase (or calibration phase), and the online data collection phase.

Once all the new RF signals are assigned to estimated locations in the online data collection phase, the task times are estimated from those locations. The results reported by Atichat (2011) indicated that the quality of the resultant estimated task times using this approach was sensitive to the *k* parameter of the kNN algorithm. Therefore, this study focuses on the pattern recognition phase by investigating the effects of several pattern recognition parameters on the accuracy of task time estimation. Different levels of these pattern recognition parameters were identified and tested via a designed experiment on offline and online datasets collected wirelessly in a simulated assembly area covered by a WSN. More specifically, this paper analyzes how using the kNN pattern recognition algorithm to classify WSN signals affects the accuracy and computational performance of task time estimation using those signals, and identifies a limitation of the kNN algorithm in detecting task transition events. The results obtained in this research show that the parameters of the kNN algorithm can be experimentally tuned to improve the accuracy of task time estimation and to dramatically reduce computational time.

The remainder of this paper is organized as follows. The rest of Section 1 describes the problem, related work and the research contribution. Section 2 presents some background on the work that motivated this research. The research methodology and experimental results are presented in Section 3. Section 4 contains a discussion of the results, and the paper ends with conclusions and recommendations for future work in Section 5.

1.1 Problem Definition

Location fingerprinting relies on the fact that an RF signal degrades (i.e., attenuates) with distance. This phenomenon, referred to as *free space loss*, is modeled by the radio propagation model which expresses the RF signal power received by an antenna as a function of the signal power transmitted by another antenna (Ahson & Ilyas, 2011; Stallings, 2005):

$$P_r = P_t \left(\frac{\lambda}{4\pi d}\right)^n G_t G_r \tag{1}$$

where P_r is the received signal power, P_t is the transmitted signal power, λ is the wavelength of the RF signal, G_t and G_r are the transmitter and receiver gains, respectively, d is the distance between the transmitter and receiver, and n is a signal path loss coefficient which is determined by the environment and typically ranges from 2 to 6. The received signal power, P_r, is generally reported by a surrogate measurement such as the link quality indicator (LQI) or the received signal strength indicator (RSSI). The strength of an RF signal transmitted from a source can then be measured (as the LQI or RSSI) at several receivers and the relative degradation in the power of the signal at each of the receivers is then used to estimate the location of the signal source relative to the receivers.

The RF signal transmitted from a source encounters several forms of interference in its path to the receiver, which affect the measured strength and stability of the signal. For example, obstacles like walls and human bodies cause the RF signal to attenuate (i.e., lose signal strength), whereas signal reflection, scattering, and refraction result in losses due to the multipath effect. Therefore, the signal measurement used in a location fingerprinting system has to be processed to compensate for the variation introduced by these sources of noise.

This variation in location accuracy is compounded when the locations classified by location fingerprinting are used to estimate the duration of a transmitted signal at each of those locations. The issue is that the key to task time estimation is the ability to accurately detect the transition from one task to the next, which is represented by the transmitted signal moving from one location to another. In an RF-based location fingerprinting system, there is no easy way to discriminate (from two consecutive signals in time) whether an apparent change in the source of the RF signal from one location to another is due to the signal transmitter moving from the first location to the next, or whether it is due to variation in a noisy signal environment.

To address the specific problem of location accuracy due to RF signal propagation variation and its resultant effect on task time estimation, the effects of various pattern recognition parameters on the accuracy of task time estimation were investigated in this research. More specifically, experiments were conducted using different levels of the kNN classification algorithm that was used in location classification (the level of the *k* parameter), alternate methods of recording the offline fingerprinting data (the fingerprinting method), different methods for calculating the proximity of a new signal to existing signals in the offline database (the distance metric), as well as alternate formulations for detecting the transition of a signal transmitter from one task (as represented by a location) to the next (the segmentation algorithm).

1.2 Related Work

Related work in the area of wireless task time estimation can be aggregated into two main areas, i.e., *work measurement* and *location fingerprinting*.

The most accurate manual work measurement methods involve time studies, but time studies are not effective at measuring the task times of non-cyclical or long tasks, like those in healthcare (Ben-Gal et al., 2010). In such situations, work sampling is the preferred work measurement method. Moreover, time studies are also subject to an “observer effect” where the time study subjects change their behavior due to being observed (Franke & Kaul, 1978). In such a situation, work sampling is again a less intrusive form of work measurement. However, work sampling often requires a large number of labor-intensive observations to meet the desired accuracy (Finkler et al., 1993).

Several examples exist in the literature where measurement and estimation techniques have been used to extract location information from RF signals. RADAR, developed at Microsoft Research, was the first RF-based technique for location estimation and user tracking (Bahl & Padmanabhan, 2000). RADAR is built on top of the IEEE 802.11 Wireless Local Area Network (WLAN) standard, commonly known as WiFi. In their experiments, the researchers used location fingerprinting to build an offline database of signal strengths received at three base stations from specific physical locations whose (x, y) coordinates they recorded. They then used a kNN algorithm to classify new signals in signal space according to the offline database and, by extension, to estimate the physical locations of the new signals. This study was one of the first to provide localization via WiFi technology, and documented the impact of node orientations, the number of sampling data points, and the fact that signal strength was a stronger indicator of location than signal-to-noise (SNR) ratio. An accuracy of 80% was achieved in location estimation with a position error smaller than three meters. However, the kNN algorithm consumed significant amounts of computing power and time, which would prevent the implementation of this technology in a real-time tracking system (Honkavirta et al., 2009). Researchers have worked on improving the RADAR approach by using more access points during fingerprinting (Honkavirta et al., 2009; Jan & Lee, 2003) and by applying additional fingerprinting methods like weighted kNN, Bayesian filtering, and Kalman filtering (Honkavirta et al., 2009).

Location fingerprinting has also been used with other RF technologies. For example, SpotOn is a three-dimensional (3D) location sensor based on radio frequency identification (RFID) technology (Hightower et al., 2000). SpotOn utilizes an RSSI distance interpolation technique which includes a unique calibration technique that results in a high precision radio map between RSSI values and the distance between an RFID reader and the tag. In the calibration phase, the custom design of the SpotON RFID device allowed the researchers to fine-tune the RF signal level for both the readers and the tags to achieve a linear relationship between distance and RSSI in the radio map. This study claimed that the system can achieve very precise 3D location accuracy within a small area. However, a complete system has not been made commercially available yet. A two-dimensional (2D) location sensor system based on SpotOn demonstrated 2D location precision of 2-8 centimeters with more than 80% accuracy in a real inventory application (Ehrenberg et al., 2007).

From the review of the work measurement literature, it is evident that the amount of data collected is one of the most important factors for both time studies and work sampling techniques. The increase in data points proportionally enhances the accuracy of the estimated time but is either expensive (time studies) or may not meet the accuracy requirements (work sampling). The location fingerprinting literature suggests wireless technologies can be used as the foundation for an automated method to collect the necessary data for work measurement applications. For example, the location of a person relative to certain areas in a workstation can be identified, so that the period of time spent by that person in these areas can be allocated to the appropriate positions.

1.3 Research Contribution

As documented above, several studies have addressed *location accuracy* of location fingerprinting systems, and have attempted to reduce the time required to convert a signal into a position. While some research has been published on identifying activities over a time horizon (Ward et al., 2006), there is no current evidence of research employing the signal strength characteristics of WSN RF signals to support task time estimation applications. Moreover, *time accuracy* of a task time estimation system based on location fingerprinting has never been addressed. It is expected that this research would fulfill this gap in the body of literature.

2. Background

The following sections briefly describe the work that generated the data set analyzed in this research.

2.1 Wireless Sensor Network Design

A WSN that used RF signals to transmit the location of an operator to beacons in a simulated assembly area was designed. The WSN employed a tree network topology consisting of one centralization node, three beacon nodes and one or two tag nodes. All six nodes were Jennic JN5139, a 2.4-GHz low power wireless microcontroller compliant with the IEEE802.15.4 ZigBee standard for wireless personal area networks (Jennic, 2008). The centralization node received the signals from the three beacon nodes and communicated with the main computer through a universal asynchronous receiver/transmitter (UART) at a data rate of 115,200 bits per second (bps).

The WSN was set up on the JenNet protocol stack. The JN5139 microcontroller measures received power in terms of a link quality indicator (LQI) value on an integer scale that ranges from 0 to 255, where 255 represents the strongest signal. The LQI value is updated every time the module receives new data packets from other nodes.

2.2 Simulated Assembly Area

A simulated assembly area was constructed that consisted of three workstations setup in an area 100 inches long by 180 inches wide. Each individual workstation was equipped with a beacon node, as depicted in Figure 1.

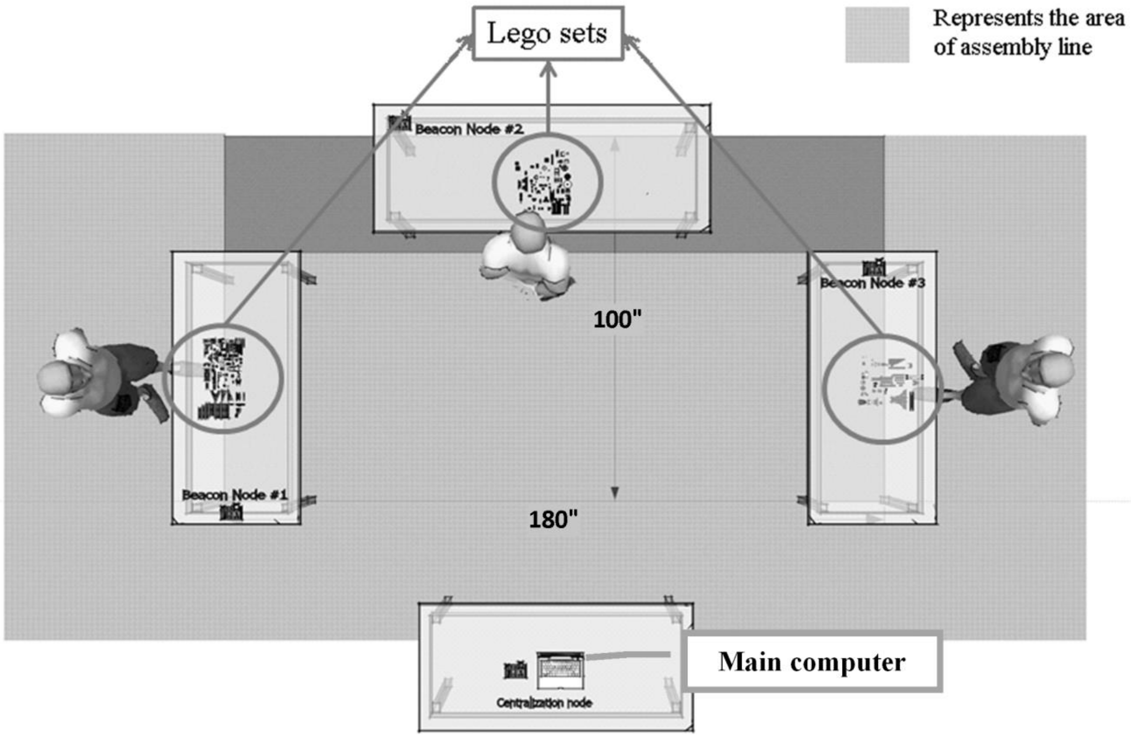

Figure 1: Layout of the simulated assembly area

The centralization node was responsible for interfacing with a data collection computer and was placed in a central location (next to the main computer in Figure 1) for best line-of-sight with the three beacon nodes. The beacon nodes in the simulated assembly area collected the LQI data from the tags nodes during the offline and online data collection phases. The tag nodes were placed on the assembly operator, on the front only or on both the front and back, depending on the experimental treatment combination. The LQI data were then used to triangulate the location of the operator in signal space. This location in signal space was then matched to a location in physical space using location fingerprinting and the kNN classification algorithm.

2.3 Location Fingerprinting

The specific method of data collection employed is referred to as location fingerprinting, and consisted of two phases:

1. ***The offline data collection phase (or calibration phase)*** where RF signals are transmitted from pre-determined locations and then stored in a database, and
2. ***The online data collection phase*** where a new RF signal is assigned to a likely location, or *classified*, based on the offline database and pattern recognition algorithms.

In these two phases, the location of the centralization node and the beacon nodes within the simulated assembly area were fixed.

2.3.1 Offline Data Collection

Since it was anticipated that the design characteristics of the WSN would influence the ability to accurately estimate individual task times, a 2^4 factorial designed experiment was conducted, resulting in 16 treatment combinations. The offline data were collected under all 16 treatment combinations. The four specific WSN design factors investigated, together with the treatment levels, are shown in Table 1.

		Factors			
		(A)	(B)	(C)	(D)
		Number of Tag Nodes	Number of sample LQI values collected per site survey grid location and orientation	Number of tag node orientations at each grid location	Number of site survey grid locations
Level	+	1	2,000	4	5
	-	2	6,000	8	9

Table 1: Experimental controlled factors

The number of tag nodes used in the offline data collection phase was either one or two. The justification for this was to investigate the effect of the tag node antenna's radiation coverage. The number of sample LQI values per site survey grid location and orientation was either 2,000 or 6,000. This enabled the investigation of the effect of sample size per location. The number of tag node orientations at each site survey grid location was either four (i.e., North, East, West and South) or eight (i.e., North, NE, East, SE, South, SW, West, and NW). Finally, the number of site survey locations per treatment was either five or nine. Figure 2 and Figure 3 show the site survey positions in the simulated assembly area utilized for the five-location treatment combinations and the nine-location treatment combinations. Table 2 describes how the site survey locations for each treatment level were assigned to one of three assembly workstations.

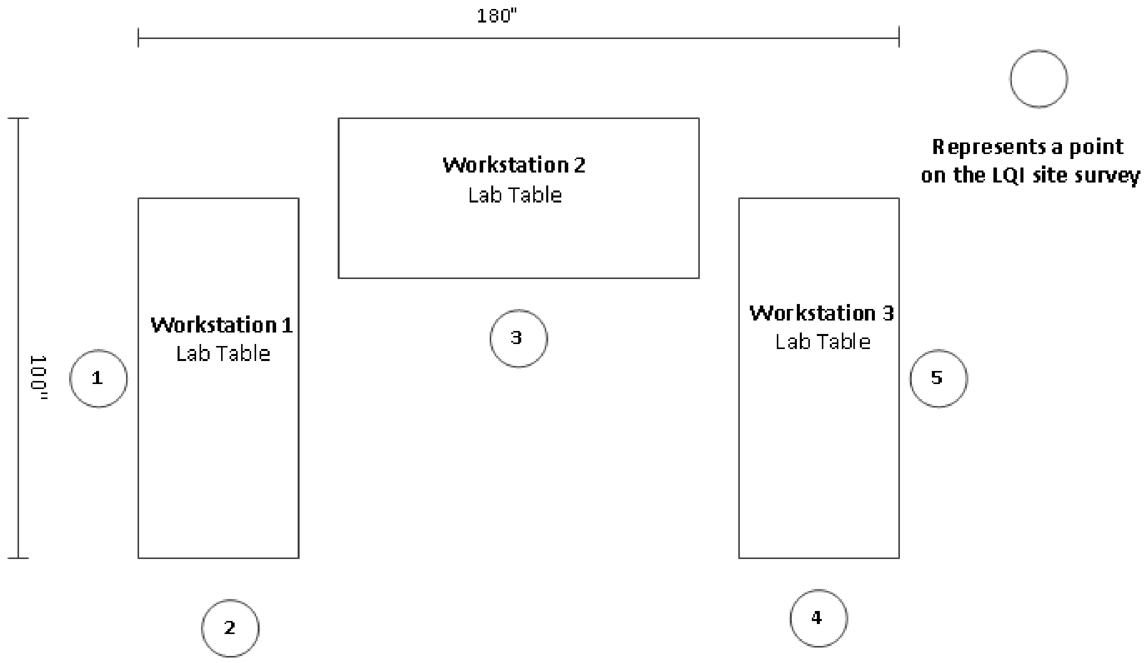

Figure 2: Positions of the five-location site survey

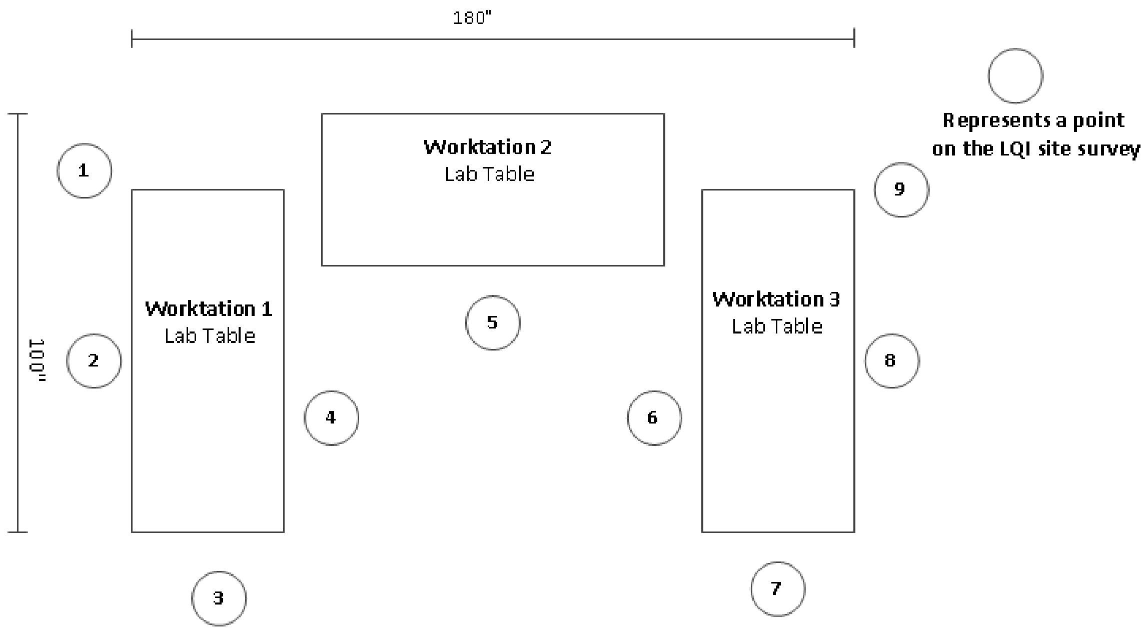

Figure 3: Positions of the nine-location site survey

Number of Site Survey Grid Locations	Locations associated with Workstation #1	Locations associated with Workstation #2	Locations associated with Workstation #3
5	1, 2	3	4, 5
9	1, 2, 3	4, 5, 6	7, 8, 9

Table 2: Relationship between site survey locations and workstations

During the offline data collection phase, a data collection software application utilized a wireless data packet counter feature which automatically forced the centralization node to stop receiving data packets once the required number of LQI sample values (i.e., 2,000 or 6,000) had been reached. The LQI values collected for each of the 16 offline templates were stored in a Microsoft® Access database.

2.3.2 Online Data Collection

The purpose of the online data collection phase was to generate LQI values that could be used to evaluate the effectiveness of each of the 16 offline templates in estimating the location of the operator within the simulated assembly area, so that individual task times could be calculated. To this end, the operator was equipped with a single tag node and allowed to move freely in the simulated assembly area to perform 20 runs of a job consisting of assembling a different Lego™ set at each of the three workstations. The Lego™ sets varied in their level of difficulty.

In each online run, the operator started the job at a randomly selected workstation. Once the first Lego™ set assembly task was completed, the operator then randomly moved to the next workstation until all the Lego™ set assembly tasks were finished.

LQI values with time stamps were automatically collected from the three beacons and sent to the main computer via the centralization node. It is important to note that the number of LQI values collected in each of the 20 runs were not always the same due to variability in the communication speed between the nodes in the network. This problem was addressed by considering only the first ten LQI values reported by each beacon node within every second for a period of five seconds. Each record consisted of a time stamp and the three LQI values sampled at that time from each of the three beacons.

The time the operator spent at each workstation during each of the 20 online runs was recorded manually using a stopwatch. The manually recorded times were then stored in a spreadsheet and this ground truth data were later used as a baseline for measuring the ability of the WSN-based task time estimation system to estimate individual task times.

2.4 Location Estimation using kNN

In order to estimate the amount of time the operator spent at each workstation of the simulated assembly area, the location of the operator had to be estimated first. This was accomplished by using the kNN algorithm to classify (based on their LQI level) the RF signals received every five seconds by the beacons during the online runs. The kNN assigns a new signal to a location as follows:

1. It compares the signal to all the reference measurements in the offline database and selects the *k* values that are closest to the one being reviewed.
2. It then performs a vote of the locations within those *k* selections. The new signal is assigned to the location with the majority vote.

For example, suppose that k=3 was used, and that the strength of a new RF signal, s_1, received by each of the three beacons from the operator's tag in the simulated assembly area was measured to have an LQI of 100. In this case, the kNN algorithm would calculate the distance (in signal space) between s_1 and all the other LQI measurements in the offline site survey database and select the three records with LQI values closest to 100 for inclusion into the majority voting step (see section 3.1.2 for details on how the distance between locations in signal space is calculated). Suppose further that two of these three records were from location A and one was from location B. Based on majority vote, the kNN algorithm would classify s_1

as belonging to location A and would predict that the signal was transmitted from location A.

2.5 Task Time Estimation

After all of the raw online RF signals were classified by location, the amount of time spent at each location was calculated. First, the total duration of a specific online run was determined by subtracting the last time stamp for that online run from the first time stamp. Next, the classified locations for that online run were grouped by workstation according to the location-to-workstation mapping shown in Table 2 and summed for each station. The amount of time the operator spent at each workstation was then estimated as a fraction of the total online run duration by dividing the number of locations assigned to a workstation by the total number of locations classified for the online run, and then multiplying this fraction with the total duration of the online run. Finally, the resulting estimated task time per workstation per online run was compared to the associated ground truth task time measurement, and the prediction error was estimated according to the following formula:

$$\%Error = \left|\frac{Observed\ task\ time_n - Estimated\ task\ time_n}{Observed\ task\ time_n}\right| \times 100 \qquad (2)$$

3. Research Methodology

Figure 4 shows the three phases of the methodology followed in this research. The methodology started with three separate data collection phases. The data collected during the *site survey* phase populated the offline database. The data collected during the *operator tasks* phase populated the online database. Finally, the data collected during the *time study* phase populated the ground truth database. The location estimation phase involved varying parameters of the kNN algorithm implemented as a Visual Basic application. The task estimation phase was also accomplished using the same custom Visual Basic application.

The next sections describe several of the focus areas of this research in more detail, including location fingerprinting, distance metric, segmentation, the kNN algorithm parameters, and the blocking factors *offline template*, *online run* and *workstation*.

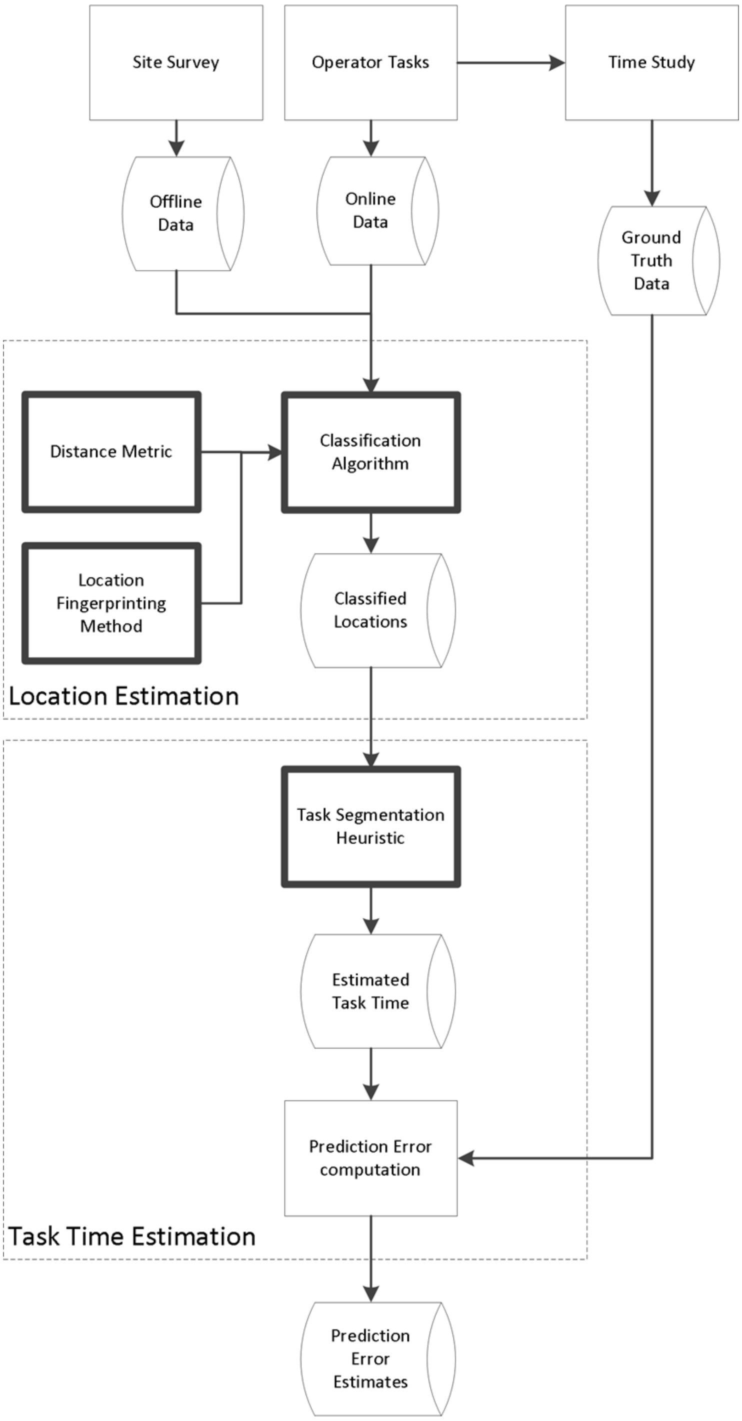

Figure 4: Process for determining the Task Time Prediction Error of different WSN designs

3.1 Pattern Recognition Main Factors

Prior results suggested that the pattern recognition method had a significant effect on the accuracy of task time estimation (Atichat, 2011). Thus, the effects that some parameters of the kNN algorithm had on the accuracy of the task time estimation were further investigated via a designed experiment. During the course of this investigation, it was realized that the method of allocating time to each workstation, or *segmentation*, also deserved further investigation. The primary response variable in the experiment was percent error of the task time estimate. A secondary response variable was the amount of computational time required to classify an online run. The motivation for investigating this secondary response was that the kNN algorithm uses the entire offline site survey database to classify one new test sample and can therefore be expensive in terms of computational resources for large site survey databases (Bishop, 2006). Table 3 shows the pattern recognition main factors (and their levels) selected for the experiment.

		Factors			
		(A)	**(B)**	**(C)**	**(E)**
		Fingerprinting	**Distance metric**	**Segmentation**	**Level of k parameter**
Level	+	Average LQI	Weighted Euclidean	Jumping Window	1
	-	Individual LQI	Euclidean	Proportional	3

Table 3: Controlled design factors for the pattern recognition experiment

3.1.1 Fingerprinting

Fingerprinting, the first experimental factor of interest, refers to whether individual LQI measurements at a survey location are used as references in the site survey database or whether a summary measure of central tendency (e.g., average, mode or median) is used instead. The original work by Atichat (2011) used individual LQI measurements for the fingerprinting data. However, exploratory data analysis of the resulting offline database revealed wide variations in the signal strength by location. These variations and the resulting overlap in received signal strength from adjacent (or even further apart) locations increased the likelihood of misclassification. Figure 5 below highlights this issue.

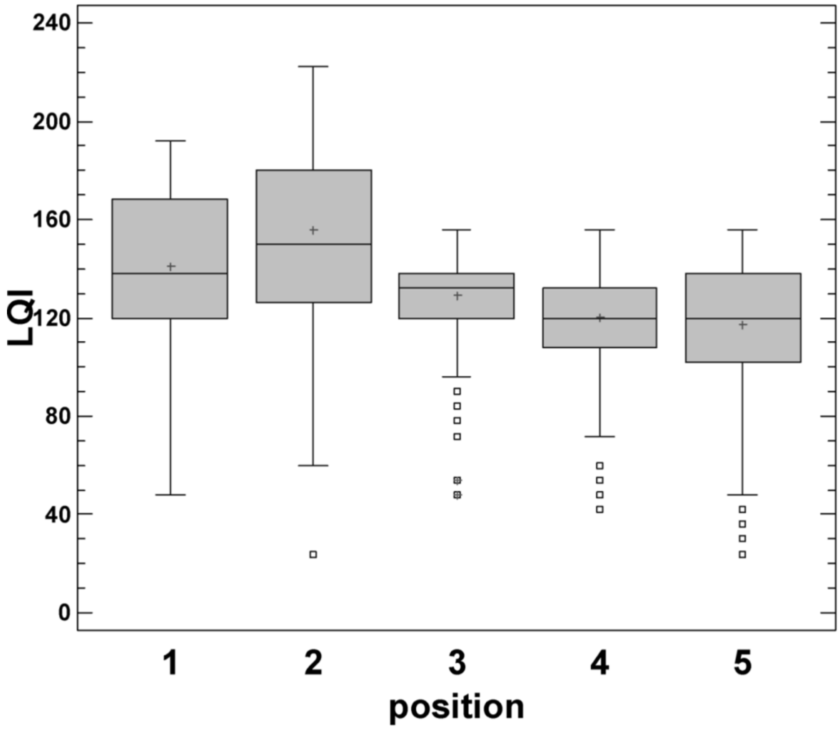

Figure 5: Variation in the Site Survey LQI data for offline template 4, beacon 1

The box-and-whiskers plot in Figure 5 aggregates all 40,000 recorded LQI data points from beacon 1 in offline template 4 (offline template 4 is the site survey treatment combination of one tag node, 2,000 LQI samples, four orientations, and five positions). The overlap in interquartile range between the positions is very evident. Yet, there is convincing evidence that the LQI values at the different locations are different (One-Way ANOVA F-test, p-value < 0.0001). The Fisher's Least Significant Difference (LSD) plot depicted in Figure 6 supports this evidence since the 95% confidence intervals of the different location means do not overlap at all. Thus, a decision was made to investigate how the task time accuracy would change if this signal strength variation was controlled by location. As Table 3 shows, the two levels used for the experimental factor *fingerprinting* were the Average LQI and the Individual LQI at a survey location.

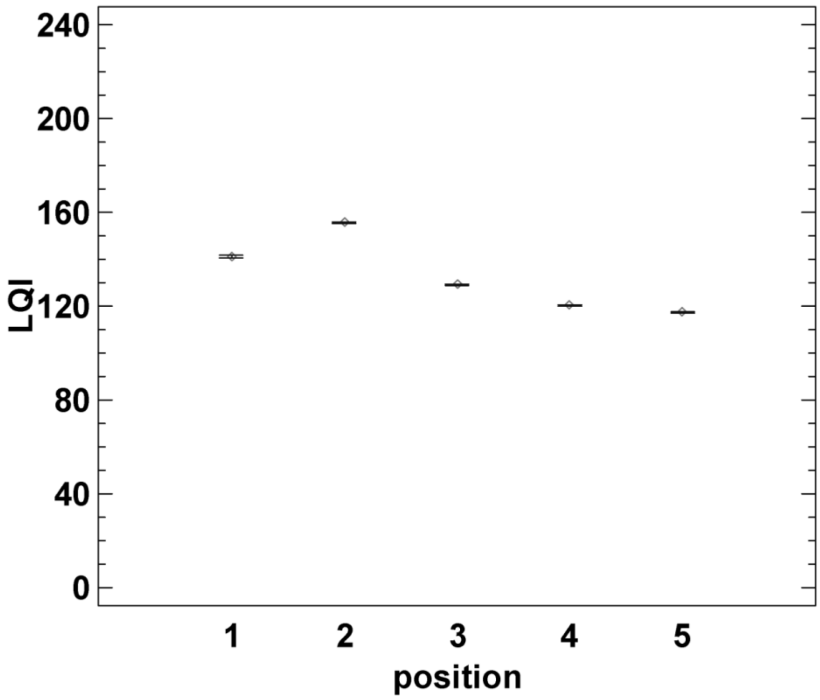

Figure 6: Fisher's LSD plot of the Average LQI for the Site Survey data for offline template 4 (beacon 1)

3.1.2 Distance metric

The second factor of interest was the distance metric used in the kNN algorithm. A distance metric is required by the kNN algorithm to calculate the proximity of a new signal to the classified signals in the offline database. Equation 3 shows the equation typically used in the kNN algorithm to calculate the Euclidean distance:

$$d(x,y) = \sqrt{\sum_{k=1}^{3}(x_k - y_k)^2} \qquad (3)$$

where d is the distance between two points x and y, each of which is described in signal space by a vector **l** = $\{l_1, l_2, l_3\}$ composed of three LQI readings from the three beacons, and k is an index to each reading in the vector **l**. However, the standard Euclidean metric assumes that each of the input variables are equally important, which gives them equal weights in the calculation of the Euclidean distance. This is not always appropriate. For example, consider Figure 5 again. If the LQI value measured by beacon 1 happens to be one of the outliers in position 5 in the box-and-whiskers plot, that LQI value will exert a disproportionate influence on the calculated Euclidean distance because of its magnitude. This will increase the likelihood of a misclassification. A method for addressing this issue is to weigh the contribution of each variable in the Euclidean distance calculation by its "relative importance" (Hand et al., 2001; Mitchell, 1997). The determination of "relative importance" is generally left to the judgment of the investigator. Thus, it was decided that the effect of penalizing signals that were far from the mean LQI received by a beacon from a location should be investigated. A way to do this is to divide each LQI value by the standard deviation of all LQI values at that location. For computational reasons, this was implemented with the variance of the signal at the site survey location, and the distance metric becomes the weighted Euclidean distance, as follows:

$$d(x, y_{kl}) = \sqrt{\sum_{k=1}^{3} w_{kl}(x_k - y_{kl})^2} \tag{4}$$

Where:

$$w_{kl} = \frac{1}{\sigma_{kl}^2}, \qquad \text{and } l = \begin{cases} 1, \ldots, 5 \text{ for five} - \text{location treatments} \\ 1, \ldots, 9 \text{ for nine} - \text{location treatments} \end{cases} \tag{5}$$

This weighted Euclidean distance was selected as a level in the distance metric factor, in addition to the standard Euclidean distance. Figure 7 illustrates the effect of this standardization of the LQI measurement on the same 40,000 data points from the template 4 site survey plotted in Figure 5 without the standardization and it can be seen that the discrimination between the five positions is magnified. The Fisher's LSD 95% confidence interval plots of the weighted Average LQI in Figure 8 further support this increased separation.

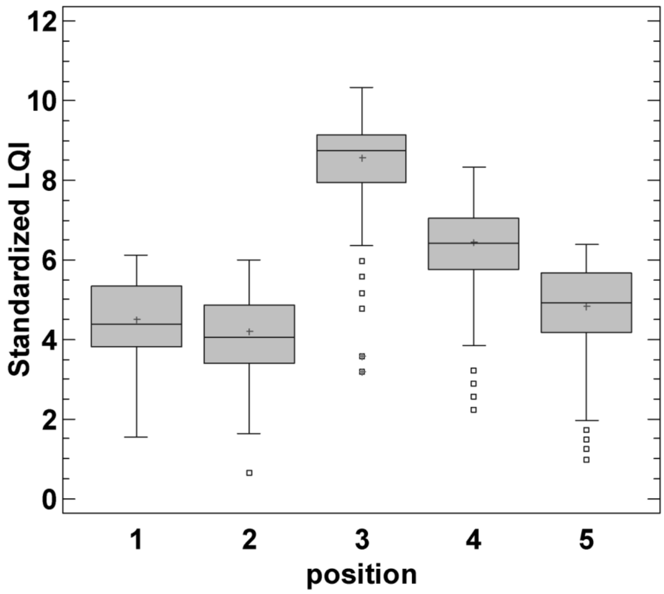

Figure 7: LQI Standardized by the Standard Deviation for offline template 4 (beacon 1)

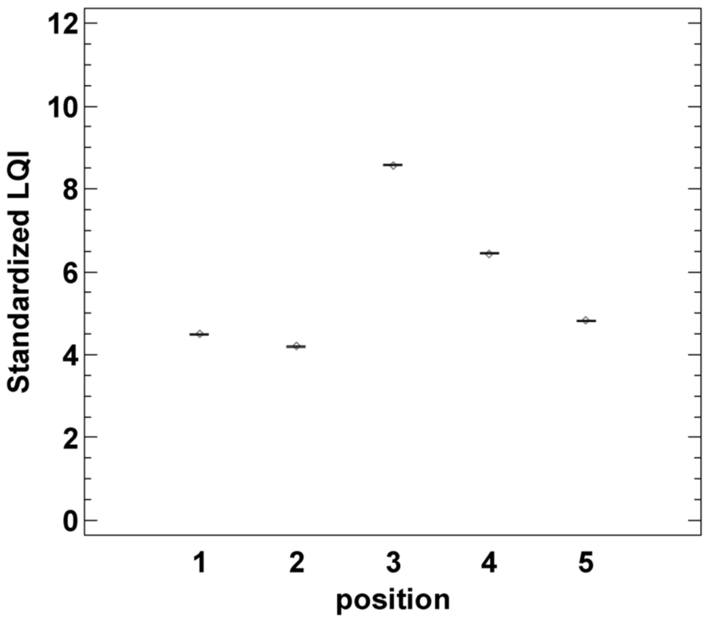

Figure 8: Fisher's LSD plots of the variance-weighted Average LQI for the standardized offline template 4, beacon 1

3.1.3 Segmentation

Segmentation, the third experimental factor of interest, refers to the algorithm used to tally the task time from individual locations. Proportional segmentation means that the task time allocated to each workstation is proportional to the fraction of locations classified by the kNN algorithm as belonging to that workstation. When the task times estimated by the proportional segmentation algorithm were analyzed, it was clear that these task times had an inherent source of error because the segmentation algorithm had no *temporal awareness*. Figure 9, Figure 10, Figure 11, Figure 12 and Figure 13 illustrate this issue for one of the 20 online runs, selected for illustrative purposes.

Figure 9 depicts the ground truth for online run 1, as measured with a stopwatch and it can be seen that the operator spent approximately 4 minutes and 15 seconds at location 3, then spent approximately 7 minutes and 55 seconds at location 1, and finally spent approximately 8 minutes and 45 seconds at location 5.

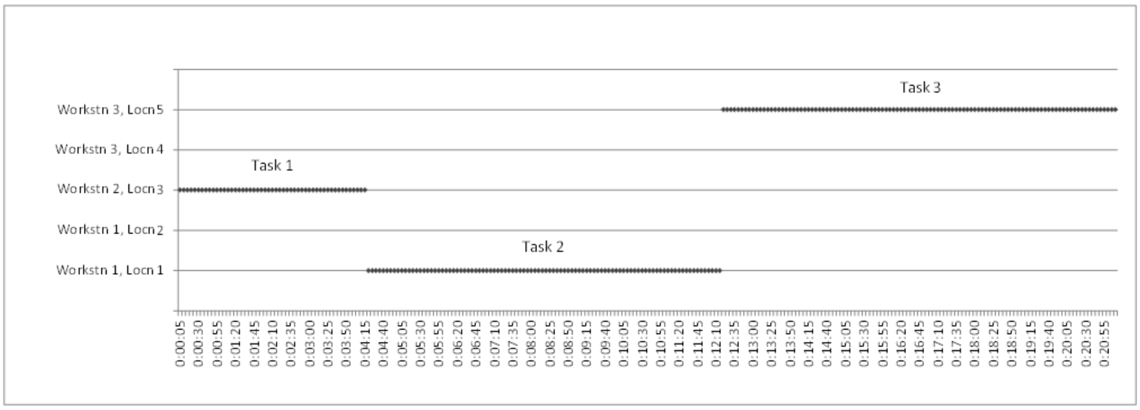

Figure 9: Online run 1 ground truth. This was the actual time study task time measured with a stop watch at each location 3 (workstation 2), location 1 (workstation 1), and location 3 (workstation 3)

Figure 10 depicts the estimated location of the operator over time after classification by the kNN algorithm (using offline template 4) and it can be seen that the kNN algorithm did an excellent job of correctly classifying when the operator was in workstation 2 performing the first task and in workstation 3 performing the third task (represented by location 3 for workstation 2, and locations 4 and 5 for workstation 3, according to Table 2). However, the kNN algorithm had trouble with correctly classifying the operator's stay in locations that corresponded to workstation 1, misclassifying roughly 50% of locations belonging to workstation 1 as belonging to workstation 2 (location 3).

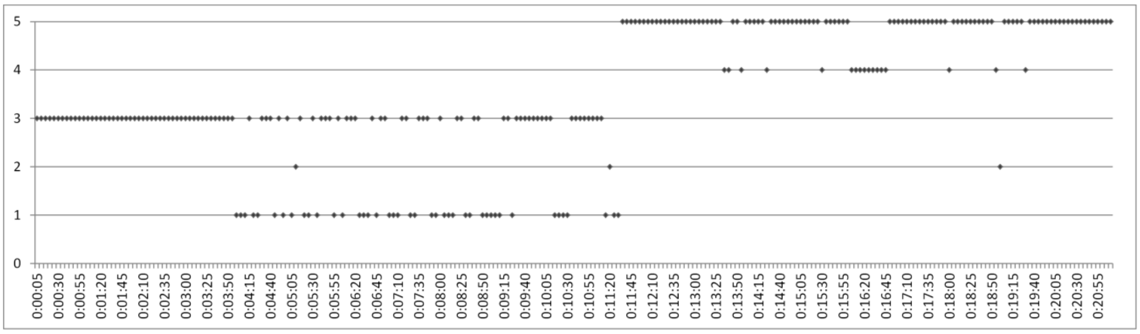

Figure 10: Online run 1 classified by the kNN algorithm using site survey template 4, k = 1.

The proportional segmentation algorithm, which follows the kNN algorithm in the pattern recognition process for this experiment, has no way of accounting for what is clearly a gross misclassification of workstation 1 locations by the kNN algorithm. These kNN misclassifications then directly translated into an incorrect prediction of estimated task times by the proportional segmentation algorithm and a resulting loss of task time accuracy. Figure 11 further emphasizes this error propagation with a simple simulation of the effect on workstation 2 task time estimation error of misclassifying workstation 1 locations by the kNN algorithm for misclassification rates from 10% to 50%. In this example, the locations at workstation 2 (and workstation 3, which is not shown for brevity) were correctly classified by the kNN algorithm (i.e., they had 0% misclassification error). The workstation 2 task time error shown in the graph in Figure 11 is entirely due to workstation 1 locations misclassified as workstation 2 locations. This task time estimation error at workstation 2 is roughly twice that at workstation 1. This amplification is due to the approximate relative magnitude between the workstation 1 task time and workstation 2 task time.

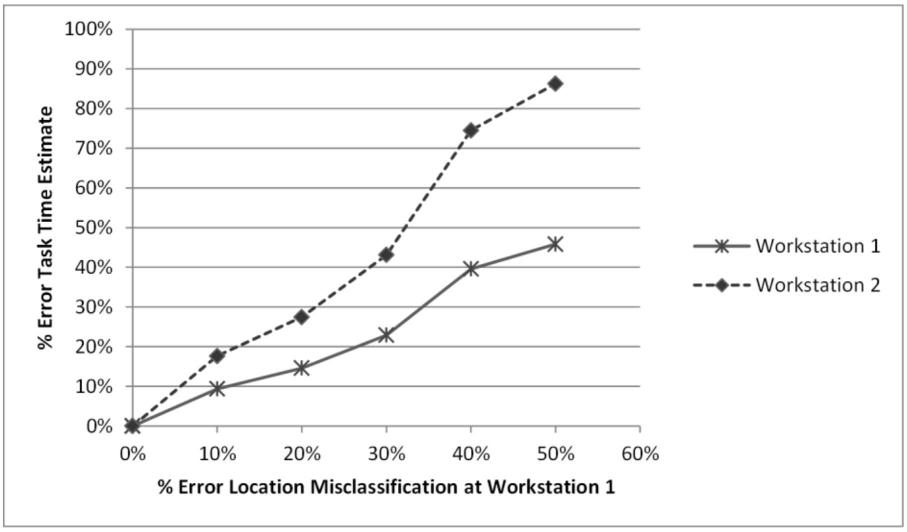

Figure 11: The effect on Workstation 2 Task Time Estimation Error of Workstation 1 locations misclassified as Workstation 2 locations by the kNN algorithm

To address this misclassification issue, a jumping window algorithm was developed which makes a second pass through the kNN algorithm classifications and classifies consecutive subgroups of the locations according to a plurality vote. The subgroup selected for this experiment was five classifications or 25-second windows (since each kNN classification represents the location of the operator in a 5-second interval). From this point on, this algorithm will be referred to as the JWMV-5 segmentation algorithm. This subgroup was selected to avoid ties (odd number of frames) and to detect a task transition within a minute of occurrence. The jumping window method is designed to reduce some of the noise in the first-pass kNN classifications. Figure 12 shows the effect of applying this algorithm to the same data set as Figure 10. Figure 13 illustrates (via five figures) how the raw LQI values are processed by the JWMV-5 segmentation algorithm and demonstrates what differentiates the task time estimation problem from a location classification problem, i.e., the need to distinguish a transition by the mobile tag from one location to another on one hand, from random signal noise on the other hand. In other words, it is a sequential data classification problem, with noisy data masking the state transitions.

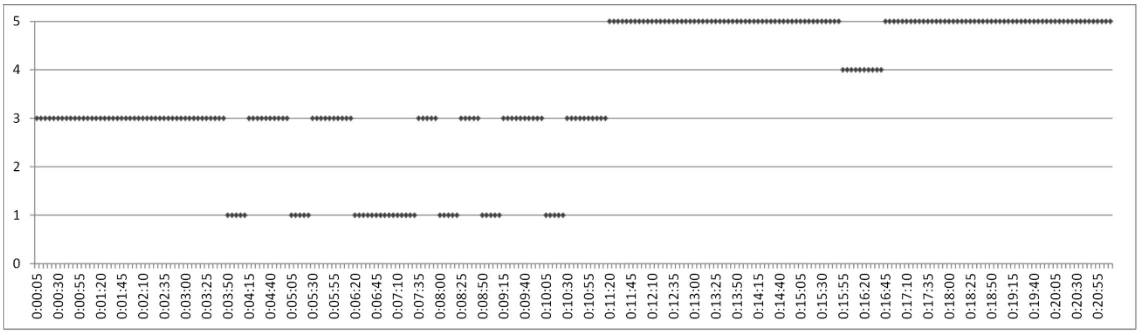

Figure 12: Online run 1 classified with the kNN algorithm using site survey 4, k = 1, and a 5-frame jumping window algorithm

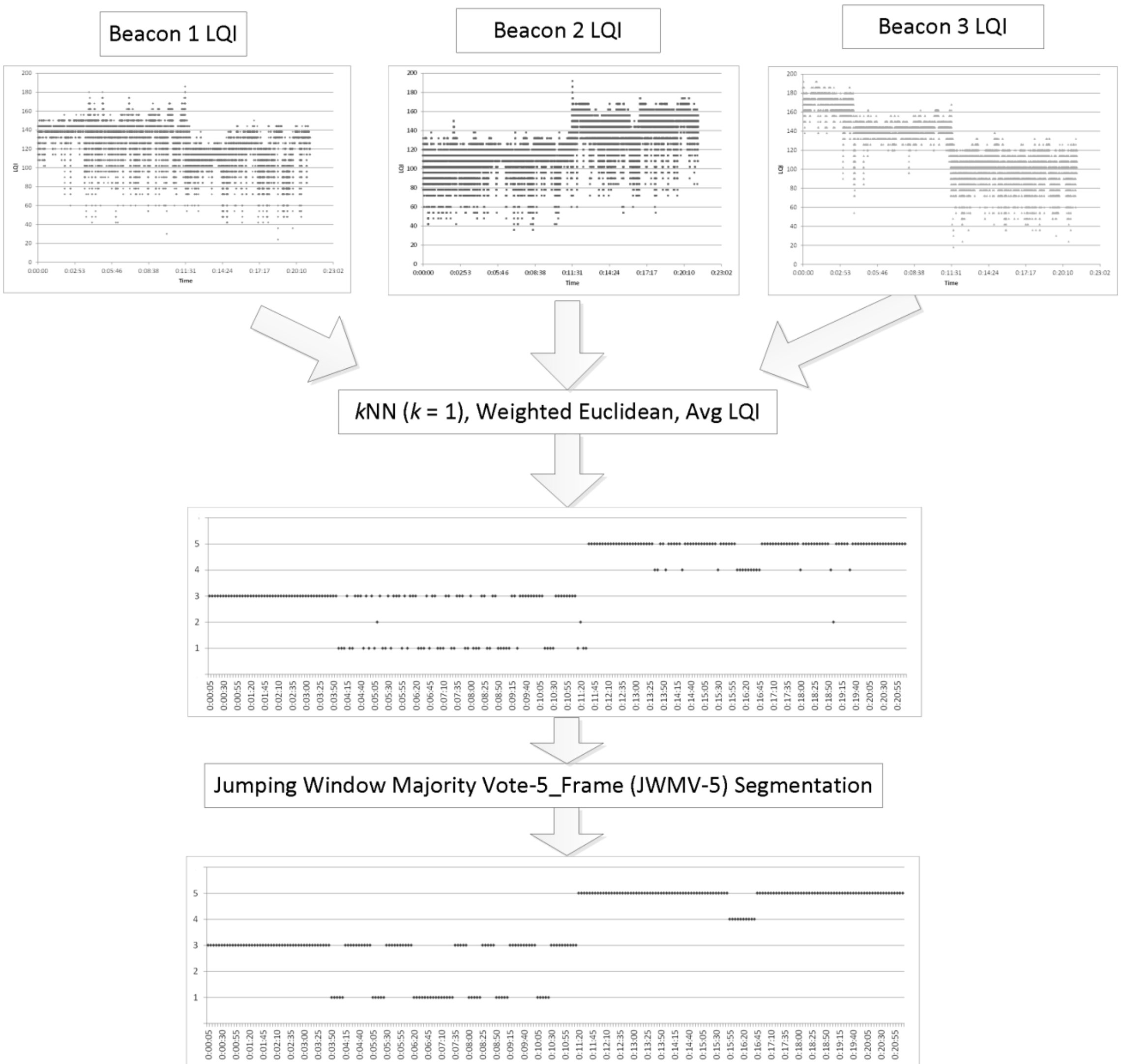

Figure 13: Overall processing of online run 1 signals, from the three LQI readings at the three beacons, to the kNN classification in Figure 10, and finally to the jumping-window processed sequence in Figure 12

3.1.4 Level of k parameter

The fourth factor of interest was the level of the *k* parameter used in the kNN algorithm. This parameter controls how many neighboring reference RF signals in the offline database are taken into account by the algorithm during the classification of a new RF signal. The values of $k = 1$ and $k = 3$ were selected for this experiment, taking into account the computational effort, the fact the Average LQI levels in the fingerprinting factor only had five or nine positions available for consideration as nearest neighbors, the desire to avoid ties (leading to a choice of odd-numbered *k*'s), and literature findings suggesting that $k = 1$ can provide good results (Ripley, 1996).

3.2 Blocking Factors

For this experiment, three blocking factors that contributed significantly to RF signal strength variation were also considered. While important for the design of the WSN in the original problem, they were not considered relevant to the pattern recognition problem. Thus, they were controlled in the experiment but their results were not treated as significant. These variables were the *offline templates*, the *number of online runs*, and the *workstations*. They are now briefly described.

Six offline templates (out of 16) were selected from the site survey database because they resulted in the best and worst task time error estimates in the prior work. This subset of the original 16 offline templates was also selected to save time since it took 464 hours to re-classify the 20 online runs using the 16 templates and the kNN algorithm for the original WSN design experiment. Table 4 below summarizes the characteristics of the selected templates. They formed the 6 levels of the categorical variable *offline template*.

Offline Template	Factors			
	(A)	(B)	(C)	(D)
	Number of Tag Nodes	Number of sample LQI values collected per grid location and orientation	Number of tag node orientations at each grid location	Number of site survey grid locations
2	2	6,000	8	5
3	1	6,000	4	5
4	1	2,000	4	5
6	2	2,000	8	5
9	1	6,000	4	9
13	1	2,000	4	9

Table 4: Characteristics of the WSN design templates selected for the kNN effectiveness study.

The next blocking factor included in the experiment was online runs. Each estimate of task time accuracy resulted from classifying one of these online runs using the kNN algorithm under the selected offline template settings. There were 20 online runs for which LQI readings and ground truth data were collected at the time the data in the offline database was collected, and it was intended that all of them would be used to evaluate pattern recognition parameters. So for this experiment, the 20 online runs were considered part of that single replication of the experiment. The online runs were also considered random factors, leading to a mixed-model experiment with both fixed effects and this one random effect (i.e., online run).

The third blocking factor was the workstation. In the work performed by Atichat (2011), this factor had the strongest effect on task time accuracy of any of the factors. So, it was decided to control for it in this experiment, using the three workstations as the three levels.

3.3 Experimental Setup

The pattern recognition experiment consisted of computer runs that estimated the task times during the 20 online runs under the treatment combinations described in Sections 3.1 and 3.2. The experiment was run on a PC with an Intel® Core ™ i7 970 3.20GHz CPU, with 16 GB of RAM, and running the Microsoft Windows 7 Professional SP1 64-bit Operating System. The offline site survey database was stored in a Microsoft Access database Version 14 (32-bit). The kNN algorithm and the task estimation algorithms were encoded in a custom Microsoft Visual Basic application, developed using Microsoft Visual Studio 2010.

3.4 Results and Analysis

The statistical analysis of the experimental results was conducted using the StatGraphics Centurion XVI statistical software. Table 5 and Table 6 show the results of the ANOVA performed on the experimental results. For brevity, only the significant two-way interactions are shown in Table 6, i.e., the table of means. As mentioned before, the *offline template*, *online run*, and *workstation* were considered blocking factors. Therefore, any effects involving those factors were ignored. However, the factor *online run* was also treated as a random effect, and because of the resulting randomization restrictions on the experiment, some of the F-ratios include this random variable in the denominator. Furthermore, as mentioned before, the data collection of the original experiment had one replication (when the offline data, the online data, and the ground truth data were collected). This means that no estimate of the pure experimental error can be made from the data. However, this issue was addressed by using the *sparsity of effects* principle (Montgomery, 2008). This principle assumes that a system is usually dominated by main effects and low-order interactions. Therefore, the results are reported for only the main effects and the two-factor interactions. The mean square errors of three-factor and higher-order interactions are pooled into the residual mean square error.

Source	*P-Value*	*F-test denominator*
MAIN EFFECTS		
A:Fingerprinting	0.0534	MS_{AF}
B:Distance metric	0.0583	MS_{BF}
C:Segmentation	**0.0006**	MS_{CF}
D:Level of k parameter	0.3258	MS_{DF}
E:Offline template	0.0000	MS_{EF}
F:Online run		
G:Workstation	0.0000	MS_{FG}
INTERACTIONS		
AB	**0.0000**	MS_E
AC	**0.0065**	MS_E
AD	0.1582	MS_E
AF	0.0000	MS_E
BC	0.6365	MS_E
BD	**0.0044**	MS_E
BF	0.0000	MS_E
CD	0.9634	MS_E
CF	0.0011	MS_E
DF	0.9784	MS_E
DG	0.0048	MS_E
Adjusted R^2	56.3091%	

Table 5: Multi-factor effect model ANOVA of estimated task time percentage errors based on kNN pattern recognition design factors

Level	*Count*	*Mean*	*Standard Error*
GRAND MEAN	5760	40.1384	0.325325
Fingerprinting			
2 (Avg LQI)	2880	36.3144	2.62588
3 (LQI)	2880	43.9625	2.62588
Distance metric			
1 (Euclidean)	2880	37.7754	1.65897
2 (Weighted Euclidean)	2880	42.5015	1.65897
Segmentation			
1 (Proportional)	2880	38.1121	0.696044
2 (JWMV-5)	2880	42.1648	0.696044
Level of k parameter			
1	2880	40.3602	0.310894
3	2880	39.9167	0.310894
Fingerprinting by Distance metric			
2 1	1440	37.3248	0.650649
2 2	1440	35.304	0.650649
3 1	1440	38.226	0.650649
3 2	1440	49.6989	0.650649
Fingerprinting by Segmentation			
2 1	1440	35.1731	0.650649
2 2	1440	37.4557	0.650649
3 1	1440	41.0511	0.650649
3 2	1440	46.8739	0.650649
Distance metric by Level of k parameter			
1 1	1440	37.0704	0.650649
1 3	1440	38.4804	0.650649
2 1	1440	43.6499	0.650649
2 3	1440	41.353	0.650649

Table 6: Table of Least Squares Means for %Error with 95.0 Percent Confidence Intervals

The following conclusions can be drawn from the ANOVA results about the impact of the main effects on the task time estimation error:

- There is convincing evidence that the *segmentation* algorithm (whether coarse-grained proportional or a more fine-grained 5-frame jumping window) has a significant effect on the accuracy of the task time estimate (F-tests, p-value << 0.01).

- There is suggestive, but inconclusive evidence that the *fingerprinting* method (whether averaged or not) and the *distance metric* (whether weighted for signal variance or not) have a significant effect on the accuracy of task time estimation (F-test, 0.05 < p-value < 0.10).
- There is no evidence that the *level of the k-parameter* levels selected for this experiment (i.e., 1 and 3) have a significant effect on the accuracy of task time estimates (F-test, p-value > 0.10).
- Finally, there is convincing evidence that the following two-factor interactions have a significant effect on the accuracy of the task time estimate (F-tests, p-value < 0.01):
  - Fingerprinting with distance metric
  - Fingerprinting with segmentation
  - Distance metric with level of the k-parameter

The scope of population inferences from these findings is limited to the offline site survey, online run, and ground truth databases used for the experiment and cannot be generalized. This is because the simulated assembly process was not randomly selected from the general population, and the model selection for the kNN algorithm did not utilize a randomization technique like cross-validation. Cross-validation separates model development and testing and seeks to find a model that minimizes error estimation (Hastie et al., 2008, sec. 7.3). Limited causal inferences can be drawn from the findings because the order in which the online runs were performed was randomized. These findings will be further interpreted in Section 4.

A second response variable of interest was the computational performance of the various kNN algorithm designs. Figure 14 shows that the LQI fingerprinting treatments took as long as 40 hours to classify the largest design template (with 480,000 site survey data points) and the predictive relationship was linear. Figure 15 shows that the average duration of the Average LQI fingerprinting treatments was about 44 minutes (0.732 hours).

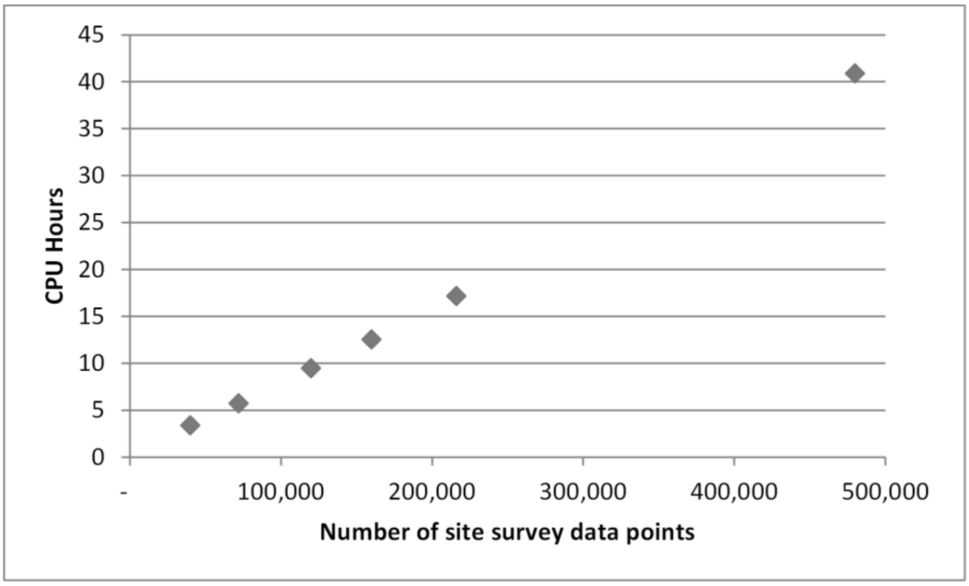

Figure 14: CPU time of the six LQI Fingerprinting splits of the experiment

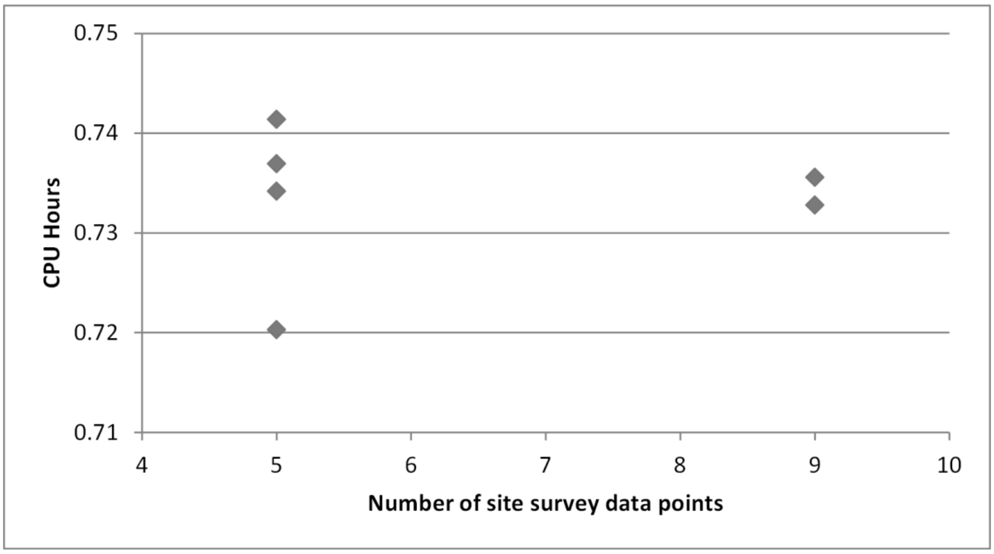

Figure 15: CPU time of the six Average LQI splits of the experiment.

4. Discussion

This section discusses and interprets the results of the experiment.

4.1 Task Time Estimation

4.1.1 Effect of Fingerprinting

There is suggestive but inconclusive evidence that the fingerprinting procedure based on Average LQI improved task time accuracy more than the fingerprinting procedure based on Individual LQI values. A possible explanation is that Individual LQI values have a higher variance per location (i.e., 2,000 or 6,000 data points versus one data point for Average LQI). The higher variance per location increases the likelihood that adjacent locations can appear to be close to the new signal being classified, and hence increases the likelihood of misclassification. A second possible explanation is that additional variation introduced by the two-tag offline templates (templates 2 and 6) masked the fingerprinting signal.

4.1.2 Distance Metric

Like the case for fingerprinting, there is suggestive but inconclusive evidence that the Euclidean distance metric improved task time accuracy more than the Weighted Euclidean distance metric used in this investigation. Again, a possible explanation is that the error introduced by the two-tag offline templates 2 and 6 masked the distance metric signal.

4.1.3 Segmentation

The proportional segmentation algorithm used in the original experiment is significantly more accurate in task time estimation than the JWMV-5 segmentation algorithm developed in this investigation. One possible explanation for this unexpected result is that the 25 second voting window of JWMV-5 was selected arbitrarily, without optimizing model selection.

4.1.4 Level of k Parameter

There is no significant difference in the task time accuracy that was attributable to the levels of the *k* parameter selected for this experiment. A possible explanation is that the parameter selection was not optimized with n-fold cross-validation.

4.1.5 Two Factor Interactions

The strongest effect in the entire experiment was the two-factor interaction between the fingerprinting method and the distance metric. Figure 16 shows the nature of this interaction. The graph also includes 95% Fisher's Least Significant Difference (LSD) confidence intervals on the % Error. The graph shows that the effect of a weighted Euclidean distance metric (category 2) is very different when used with Average LQI fingerprinting (category 2) than when used with Individual LQI fingerprinting (category 3). It is estimated that the difference in % Error task time estimation between the Euclidean and weighted Euclidean is 2% at the Average LQI level (95% C.I.'s 0.7% to 3.3%) and is -11.5% at the Individual LQI level (95% C.I.'s -12.7% to -10.2%).

These results, together with the much faster computational time of the Average LQI splits in this experiment, indicate that in order to improve the task time estimation accuracy under the simulated assembly process conditions, the kNN algorithm should be used in conjunction with an Average LQI location fingerprinting method and a weighted Euclidean distance metric.

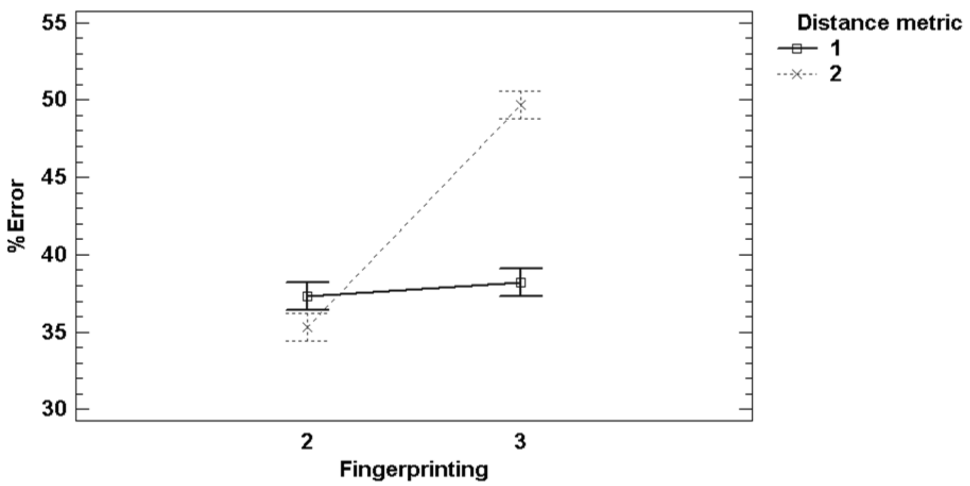

Figure 16: Fingerprinting by Distance metric interaction

Figure 17 shows the effect of the two-factor interaction between fingerprinting and segmentation and indicates that the fingerprinting procedure that employed Average LQI (category 2) improves task time accuracy for both proportional segmentation (category 1) and JWMV-5 segmentation (category 2).

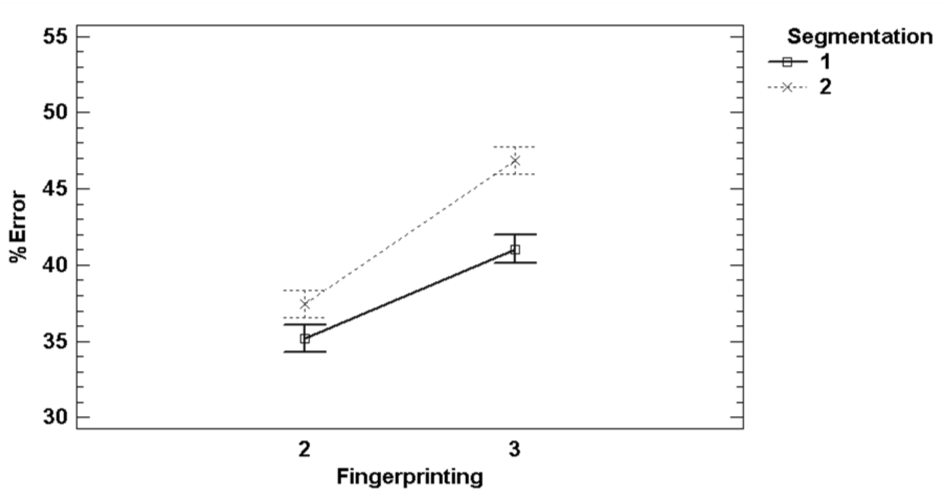

Figure 17: Fingerprinting by Segmentation interaction

The third significant two-factor interaction in this experiment was that between the level of *k* parameter and the distance metric. Figure 18 shows the nature of this interaction.

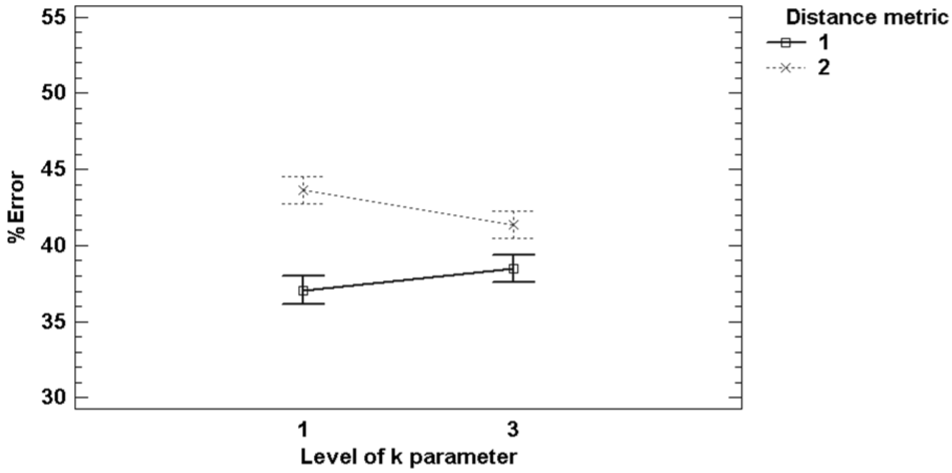

Figure 18: Level of *k* parameter by Distance metric interaction

For these limited model parameters, increasing the level of *k* improves task time estimation accuracy when used with a weighted Euclidean distance metric (category 2), but it makes the task time estimation accuracy worse when used with the Euclidean distance metric (category 1). A plausible explanation is the possibility of a bias-variance trade-off. The lower level of *k* (*k* = 1) increases model complexity which in turn increases the possibility of overfitting. However, overfitting also results in increased variance. This is because the kNN algorithm with *k* = 1 is capable of fitting a model where every data point in the offline database is its own class. In effect, each data point can be a model parameter, resulting in the most complex model possible given the offline database. However, any online data points that do not coincide exactly with the points used to fit the model would result in variation. Furthermore, since the chances of the LQI values of the online data point coinciding exactly with the offline database LQI values is fairly low, this would result in increased deviation from the predicted locations, and hence increased variance. With the weighted Euclidean metric (category 2), the effect of LQI variation at a location has been normalized, so the overall variance in the model is greatly reduced and the task time estimation error is generally reduced. With the Euclidean distance metric, the LQI variation at a location is higher, and this higher variance, in combination with the increase likelihood of overfitting due to $k = 1$, could result in a higher overall experimental error and worse task time estimation accuracy.

4.2 Computational Performance

The computational performance of the kNN algorithm in this experiment was dramatically different between the Individual LQI and the Average LQI splits. For the Individual LQI splits, the CPU time was a function of the number of site survey points in the classification problem (see Figure 14). For the Average LQI splits, the CPU time was practically constant, regardless of the number of site survey point (see Figure 15). The Average LQI runs were much faster, completing in an average of 45 minutes, whereas the Individual LQI runs ranged from 3.36 hours to 40 hours.

5. *Conclusions and Future Work*

The results obtained in this research show that the parameters of the kNN algorithm can be experimentally tuned to improve the accuracy of task time estimation and to dramatically reduce computational time. More specifically, replacing LQI fingerprinting and the Euclidean distance metric with Average LQI fingerprinting and a weighted Euclidean distance metric, respectively, will significantly improve the accuracy of the estimates and reduce the computational effort from days to minutes.

Additional significant effects that improved the accuracy of task time estimation included using Average LQI fingerprinting in conjunction with proportional segmentation, and using the Euclidean distance metric in conjunction with $k = 1$ in the kNN algorithm for location estimation. These findings should be further investigated with more robust model parameter estimation for the k parameter in the kNN.

A third conclusion regards task time segmentation. The combination of the kNN algorithm and the proportional segmentation algorithm investigated in this research were not very effective in detecting transitions from one task to the next, due to kNN misclassification errors and a lack of temporal awareness in the proportional segmentation algorithm. The attempt to introduce temporal awareness to task segmentation with the JWMV-5 segmentation algorithm was not effective, as it performed significantly worse on task time estimation accuracy than the baseline proportional segmentation algorithm. Further investigation should be conducted on the optimal selection of the frame window size in the JWMV algorithm. More broadly, several other pattern recognition algorithms like clustering, decision trees, neural networks and support vector machines are available in the machine learning domain and should be investigated for their effectiveness in accurately estimating the task time derived from location fingerprinting. Some techniques such as Hidden Markov Models, even have explicit temporal awareness (Bishop, 2006; Hand et al., 2001; Hastie et al., 2008; Mitchell, 1997). Incorporating least-squares or maximum likelihood techniques could also lead to estimates of the durations of sub-tasks within the measured tasks (Kim et al., 2008).

Additionally, this research was conducted using classical design of experiments and statistical inference, to provide continuity with the WSN experiment that was the original motivation for the research. Modern pattern recognition research and practice primarily uses Bayesian inference. Bayesian methods enable a researcher to make hypotheses about a *prior* governing distribution with little or no data, then adapt those hypotheses based on new data as it is collected, resulting in a *posterior* distribution that is *conditional* on the newly acquired data. Many efficient pattern recognition algorithms use Bayesian methods, with validated inference tools. As research into the use of pattern recognition algorithms in the estimation of task time is extended to include the full range of machine learning methods, the Bayesian approach to statistical inference will be more appropriate than the classical, or *frequentist*, approach.

Finally, the results obtained in this research clearly show that task time estimation based on the monitoring of the strength of RF signals within a WSN is viable and with several positive managerial implications. First, it can lead to the improvement in the accuracy of time studies in non-cyclical or long tasks where traditional manual time studies and work sampling are not accurate enough due to limited data collection. Most importantly, it can lead to a reduction in the cost of time studies. In fact, the true costs and trade-offs involved in using manual work measurements can easily be underestimated. For example, Finkler et al. (1993) found in one study that work sampling results differed from time study results (considered the “ground truth” reference) by 20% or more in eight out of 10 activities. Such a large error can have a significant impact on standard costs. On the other hand, the equivalent time study required 22 data collectors to measure the activities of eight subjects for a total of 13,000 minutes. Sustaining such a time study would be cost-prohibitive. These observations about manual work measurement methods drove the motivation to investigate automated methods for work measurement.

References

Ahson, S., & Ilyas, M. (2011). *Location-Based Services Handbook*. CRC Press.

Atichat, T. (2011, June 7). *A Wireless Sensor Network Approach for Estimating Individual Task Time* (M.S.). Oregon State University, Corvallis, Oregon.

Bahl, P., & Padmanabhan, V. N. (2000). RADAR: An in-building RF-based user location and tracking system. *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* (Vol. 2, pp. 775–784).

Ben-Gal, I., Wangenheim, M., & Shtub, A. (2010). A new standardization model for physician staffing at hospitals. *International Journal of Productivity and Performance Management*, *59*(8), 769–791.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer Verlag.

Ehrenberg, I., Floerkemeier, C., & Sarma, S. (2007). Inventory management with an RFID-equipped mobile robot. *Automation Science and Engineering, 2007. CASE 2007. IEEE International Conference on* (pp. 1020–1026).

Finkler, S. A., Knickman, J. R., Hendrickson, G., Lipkin Jr, M., & Thompson, W. G. (1993). A comparison of work-sampling and time-and-motion techniques for studies in health services research. *Health services research*, *28*(5), 577.

Franke, R. H., & Kaul, J. D. (1978). The Hawthorne Experiments: First Statistical Interpretation. *American Sociological Review, 43*(5), 623–643.

Hand, D., Mannila, H., & Smyth, P. (2001). *Principles of Data Mining*. Massachusetts Institute of Technology.

Hastie, T., Tibshirani, R., & Friedman, J. (2008). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Retrieved from http://www-stat.stanford.edu/~tibs/ElemStatLearn/

Hightower, J., Want, R., & Borriello, G. (2000). SpotON: An indoor 3D location sensing technology based on RF signal strength. *UW CSE 00-02-02, University of Washington, Department of Computer Science and Engineering, Seattle, WA*.

Honkavirta, V., Perala, T., Ali-Loytty, S., & Piché, R. (2009). A comparative survey of WLAN location fingerprinting methods. *Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on* (pp. 243–251).

Jan, R. H., & Lee, Y. R. (2003). An indoor geolocation system for wireless LANs. *Parallel Processing Workshops, 2003. Proceedings. 2003 International Conference on* (pp. 29–34).

Jennic. (2008). Product Brief – JN5139-xxx-Myy. Retrieved from http://www.jennic.com/files/product_briefs/JN5139-MO-PB_v1.3.pdf

Kim, D. S., Porter, J. D., & Buddhakulsomsiri, J. (2008). Task time estimation in a multi-product manually operated workstation. *International Journal of Production Economics*, *114*(1), 239–251.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.

Montgomery, D. C. (2008). *Design and Analysis of Experiments* (7th ed.). John Wiley & Sons Inc.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*. Cambridge University Press.

Stallings, W. (2005). *Wireless Communications and Networks* (2nd ed.). Upper Saddle River, NJ: Pearson Prentice Hall.

Ward, J. A., Lukowicz, P., Troster, G., & Starner, T. E. (2006). Activity recognition of assembly tasks using body-worn microphones and accelerometers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *28*(10), 1553–1567.