Active sampling to increase the battery life of mosquito-detecting sensor networks

Adam Cobb, Stephen Roberts, Davide Zilli
Dept. of Engineering
University of Oxford

Abstract

There is a need for collecting more comprehensive data on mosquito populations in malaria-affected countries. A better understanding of the prevalence of mosquitoes and their geographic locations would provide information that could be used in the battle against malaria. Current techniques for gathering data consist of taking readings at very localised sites and counting the number of outbreaks of the disease in different areas. This paper looks at modelling a network of sensors deployed to detect mosquitoes and produce a prediction of the local mosquito population. The compromise between battery life and the quality of the measurements is shown along with considerations over the sensor network infrastructure.

1 Introduction

The increasing prevalence of deploying small wireless devices brings the opportunity of gathering large amounts of data about our environment that was previously not possible. However, this also brings with it the requirement of implementing smart algorithms that enable these networks of sensors to adapt their behaviour to make intelligent decisions based on their environment.

Relevant work that has looked into smart sensor networks include Farinelli, Rogers and Jennings (2008) [1] and Rogers David and Jennings (2005) [2]. Focussing on wide-area surveillance, Farinelli et al. (2008) describe using techniques such as the max-sum algorithm to determine the sense/sleep schedules of sensors to maximise network efficiency. Looking at sensor networks from an agent-based game theoretic point of view, Rogers et al. (2005) introduces a sensor network where each sensor follows a static strategy to maximise a reward. This static strategy aims to maximise the overall objective of the network by allowing each device to act in its own interest. This is part of an emerging field called mechanism design [3] that combines multi-agent systems with game theory and emphasises decentralising the control of networks by allowing the agents to act for themselves.

Our interest in this area has arisen from the challenge of collecting data on mosquito populations as part of the HumBug project [4]. There is a great need to gather this information as previous attempts to map mosquito occurences geographically have either

been heavily reliant on expert opinion or have understandably been biased towards areas with evidence of malaria being transmitted [5]. Therefore the aim of the HumBug project is to distribute portable devices in malaria-stricken countries to build a more comprehensive data set of mosquito populations. This is with objective of understanding the possible causes of malaria outbreaks by combining gathered data from sensors along with our knowledge of the surrounding environment. Combining factors such as the local weather, fauna and flora with collected mosquito data could point to significant indicators of mosquito prevalence and enable preventative action to be taken in areas recommended by an overall model.

The reliance of future models on collected mosquito data highlights the importance of developing a network to cover as large an area as possible for as long a time period as possible. This paper specifically looks into the requirement of conserving battery life whilst ensuring that the quality of the overall population model is kept above a suitable threshold level.

Section 2 describes the sensor network model and also introduces Gaussian processes and Variational Bayesian linear regression. These techniques are then put together in section 3, which describes the application of active sampling. The results and conclusions of the active sampling are then displayed in sections 4 and 5 respectively.

2 Modelling the sensor network

This section introduces the setup of the sensor network. The experiment is set up as a unit square with sensors spread across in a grid-like pattern (see figure 1). The mosquito data is generated under the assumption that the insects have peak times of activity at both 6:00 and 18:00. During the daytime, this activity level is taken to be a flat low nominal level, whereas the nightime level is taken as a flat higher level. This approximated mosquito activity function gives a time dependent rate that can be used as the parameter to generate Poisson-distributed random variables corresponding to the number of mosquitoes appearing at different times. Calling each appearance of a mosquito an event, the location of each event can be sampled from a two-dimensional Gaussian distribution. In the case of figure 1, the event coordinates are drawn alternatively from two Gaussians, approximating two swarms. This time-dependent activity function to simulate the data follows page 669 of [6], which looks into the daily cycle of mosquitoes.

Apart from making an assumption about the form of the mosquito data, we also enable the sensors to have access to the true time and their location. Access to other significant factors, such as the local weather conditions, have not been included in this paper but must be considered in future analysis.

2.1 Reconstructing the mosquito density function using a Gaussian process

The limited number of devices spread over the total area requires a model that is able to take advantage of a small number of training points. Therefore a two-dimensional Gaussian process (GP) is used to reconstruct the mosquito density function as a GP is able to utilise a sparse number of input measurements to produce a model.

A two-dimensional GP is used to reconstruct the mosquito density function. The input space of the GP consists of the x and y coordinates of each sensor, along with their hourly readings. The readings must be kept positive as it is not possible to see a negative number of mosquitoes. Applying the logarithm to these readings before passing them into the GP

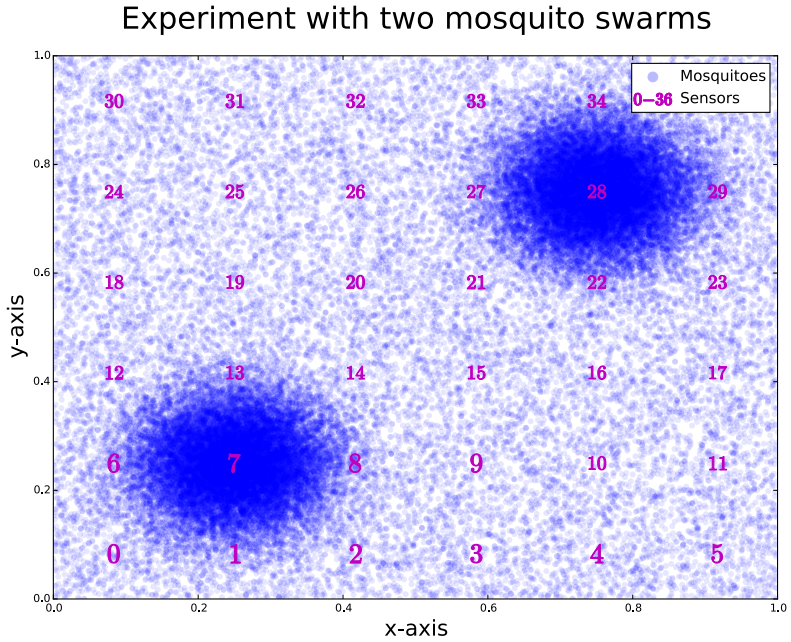


Figure 1: An example of a configuration of the mosquitoes and the sensors. There are 36 sensors spread in a grid and two swarms centred at locations $(0.25, 0.25)$ and $(0.75, 0.75)$.

enforces this positive requirement.

Building our prior knowledge into the key components of the GP is important for ensuring that the approximation is as close as possible to the true distribution. Therefore using a zero mean function appropriately accounts for the assumption that the sensors see few mosquitoes most of the time, apart from the sensors that find themselves in the vicinity of swarms (see figure 1). As the logarithm of the readings is taken, any sensor that does not detect any events in an hour of readings must be incremented to one to allow the use of the logarithm.

The covariance function defines the correlation between locations in the input space given by a kernel. Equation 1 shows the chosen covariance kernel that consists of a two dimensional Matérn $^3/_2$ kernel [7].

$$\mathbf{K}(\mathbf{r}) = \sigma^2 \left(1 + \sqrt{3}\mathbf{r}\right) \exp\left(\sqrt{3}\mathbf{r}\right) + \sigma_n^2 \mathbf{I}, \tag{1}$$

$$\text{where } \mathbf{r} = \sqrt{\sum_{d=1}^{2} \frac{\left(\mathbf{x^d} - \mathbf{x'^d}\right)^2}{l_d^2}}.$$

The d corresponds to a particular dimension of the input vector $\mathbf{x}$. The hyperparameters are l_d, σ and σ_n. σ is the output scale length and σ_n accounts for the noise as well as improving the condition of the covariance matrix. The input scale length, l_d, specifies how strong the data is correlated in each of the d dimensions. Note that $\mathbf{I}$ is the identity matrix.

This GP can then be used to produce the mosquito density function from the simulated data. Figure 2 is an example of the GP approximating two swarms of mosquitoes. The two peaks can clearly be seen as areas of high mosquito activity.

Having inferred the posterior mean surface, the total population can be estimated via an integration of this two-dimensional posterior mean over the unit square. However before

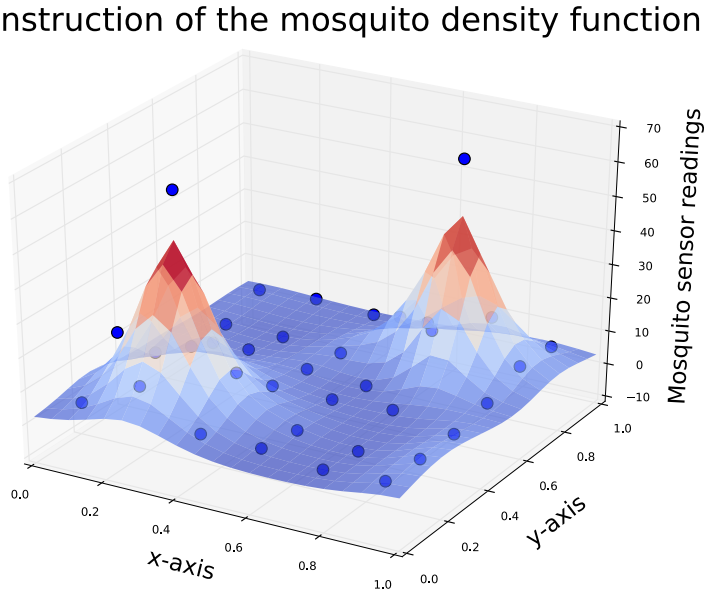


Figure 2: The GP posterior mean surface used to model the mosquito activity. The 36 sensor configuration is shown in figure 1. The sensor readings are the blue dots and the two peaks correspond to the two mosquito swarms.

integrating, the earlier normalisation step along with the application of the logarithm must be reversed. As each sensor measurement represents the local surrounding area, these measurements are in fact density readings. Therefore in order to achieve the appropriate population estimate each sensor reading is divided by its corresponding detection area, A_{det}. This division scales the double integral and is shown in equation 2. Following the same procedure, the upper and lower 95% confidence bounds are also integrated in the same way. The conversion from a log scale extends the upper bound further away from the posterior mean than the lower bound and puts a higher probability on the expectation of false positives. This higher expectation is preferred to false negatives and is aligned with the objective of the acoustic detection algorithm that is being developed to prefer false positives over false negatives.

$$pop(t) = \frac{1}{A_{det}} \int_{Area} f\left(\mathbf{x}\right) d\mathbf{x}. \tag{2}$$

2.2 Variational Bayesian linear regression with automatic relevance determination

The method of Variational Bayesian linear regression with automatic relevance determination can be used to infer a model that can be used to make predictions on mosquito population values. Roberts, McQuillan, Reece and Aigrain (2013) [8] use this technique as part of processing Kepler data sets to remove underlying trends in light curve data. Applying Variational Bayes (VB) linear regression with automatic relevance determination (ARD) requires an initial selection of basis functions. This initial selection is then reduced to the most significant set due to the ARD. The linear model is built from the basis functions, $\boldsymbol{\phi}(\mathbf{x})$, where $\mathbf{x}$ corresponds to the input data points and the weights w_n. The model (equation 3) also includes noise through the addition of a noise term $\mathbf{z}$, which

is distributed according to the multivariate Gaussian distribution, $\mathcal{N}\left(\mathbf{0}, \beta^{-1}\mathbf{I}\right)$.

$$\mathbf{t} = \sum_{n=1}^{N} w_n \boldsymbol{\phi}(\mathbf{x}) + \mathbf{z}. \tag{3}$$

The weight of each basis function, w_n, is given a Gaussian prior with zero mean and its own precision α_n. The precisions of weights that do not describe the data will become extremely large, effectively pushing the weight to its zero mean. In comparison the weights corresponding to significant basis functions will have precisions that tend to larger value. Therefore we end up with a much smaller set of significant basis functions. This resulting shrinkage is often known as ARD (Bishop 2006 [9]).

Taken from the appendix of Roberts et al. (2013) [8], equations 4, 5 and 6 demonstrate the main building blocks for VB linear regression. The aim is to estimate the posterior of the model parameters, $\boldsymbol{\theta}$, given the data, $\mathbf{t}$. The expectation of the predictive distribution along with its variance can then be calculated with respect to this posterior distribution to make a prediction. This posterior distribution is given through Bayes theorem:

$$p\left(\boldsymbol{\theta}|\mathbf{t}\right) = \frac{p\left(\mathbf{t}|\boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}\right)}{p\left(\mathbf{t}\right)}. \tag{4}$$

The vector $\boldsymbol{\theta}$ contains the parameters and hyperparameters of the model $\boldsymbol{\alpha}$, β and $\mathbf{w}$. The vectors $\mathbf{w}$ and $\boldsymbol{\alpha}$ are contain the basis function weights, w_n and precisions, α_n respectively. The likelihood, $p\left(\mathbf{t}|\boldsymbol{\theta}\right)$, describes the noise model of the linear regression model, which is the Gaussian distribution $\prod_{n=1}^{N} \mathcal{N}\left(t_n|\mathbf{w}^T\boldsymbol{\phi}_n, \beta^{-1}\right)$. The prior for $\boldsymbol{\theta}$ is given by

$$p\left(\boldsymbol{\theta}\right) = p\left(\mathbf{w}|\boldsymbol{\alpha}\right) p\left(\boldsymbol{\alpha}\right) p\left(\beta\right). \tag{5}$$

The prior on each weight, w_n, is given by the Gaussian distribution, $\mathcal{N}\left(w_n|0, \alpha_n^{-1}\right)$ and both $\boldsymbol{\alpha}$ and β are given by Gamma distributions as described in [8].

The final probability distribution to define in equation 4 is the evidence

$$p\left(\mathbf{t}\right) = \int p\left(\mathbf{t}|\boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}\right) d\boldsymbol{\theta}. \tag{6}$$

The evidence gives a measure of how likely the data is, given the parameters. It can be reformulated as a free energy term and the Kullback-Leibler (KL) divergence. This reformulation is demonstrated in [8][Eq. B1 - B6]. The KL divergence is always positive which means that the free energy term is a strict lower bound on the evidence. In VB the posterior distribution $p(\boldsymbol{\theta}|\mathbf{t})$ is approximated by $q(\boldsymbol{\theta}|\mathbf{t})$. This approximation makes the assumption that the posterior is separable, built up from the product of independent posteriors over each of the parameters, enabling the problem to become tractable.

The aim is to update the parameters of the posterior distribution by maximising the lower bound on the evidence. Noticing that the form of the free energy contains a negative KL divergence, updating the parameters of the model comes from taking the expectation of the log of the joint distribution with respect to all the other parameters.

Figure 3 demonstrates VB linear regression with ARD using a set of basis functions consisting of harmonics and polynomials. The growing confidence bounds outside the range of data provide a good indicator of our increasing uncertainty in the model's prediction.

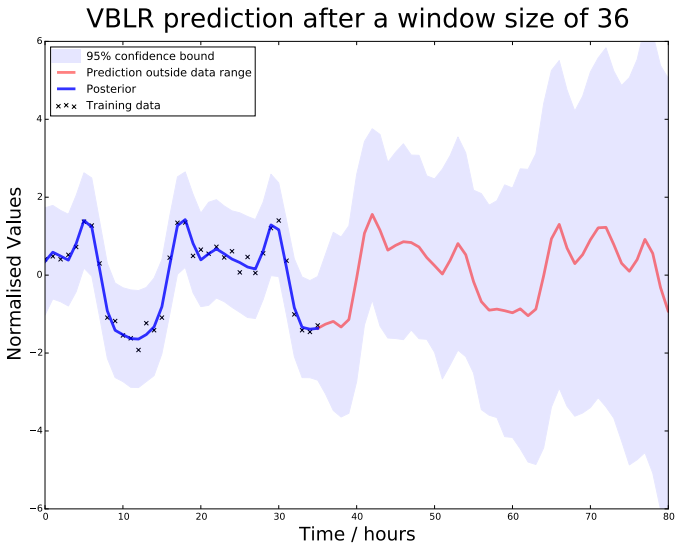


Figure 3: VB linear regression with ARD. The dark blue line corresponds to regression over the training data and the red line corresponds to mosquito population predictions at future hours. Note that the data has been normalised to have a mean of zero and a standard deviation of one.

3 Active Sampling

When a sensor takes a reading, there is a cost associated due to its battery usage. The inclusion of this constraint is covered in this section by looking at the technique of active sampling. Active sampling is the process of automatically sampling according to the network's uncertainty in its predictions. When the network's level of uncertainty in its predictions reaches a certain level, the algorithm takes a new sample. The idea to use active sampling follows from Osborne, Roberts, Rogers, Ramchurn and Jennings (2008) [10], as they successfully tested active sampling on a network of weather stations to intelligently select samples from the stations. This technique can implemented on the mosquito sensor network to reduce the battery usage. Using the VB model, population predictions are made that can be qualified by their confidence bounds. Setting a threshold on these confidence bounds and sampling according to this threshold results in active sampling.

Figures 4 and 5 show the application of this method. An initial 'burn-in' time of 8 samples is empirically chosen as it enables enough data to be collected to make predictions. Without these initial few samples, the confidence bounds never reach the set thresholds. Preventing the bounds from growing exceedingly large outside the range of the data comes from fusing the predicted posterior mean with the prior mean and variance of the data. An empirical prior of the data set variance is calculated from the training data and this prior is fused with the posterior variance from the VB model, causing the predictions to collapse to the prior once the posterior bounds get too large.

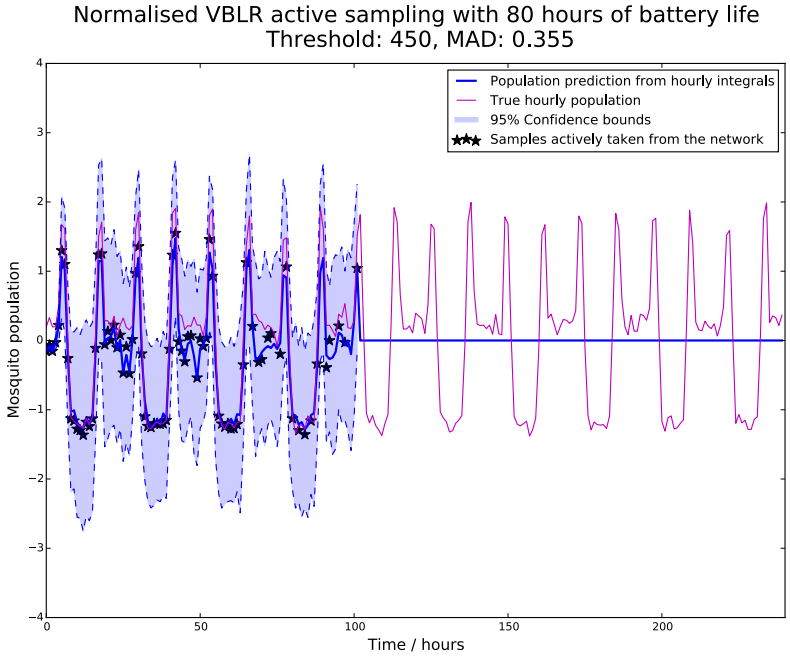


Figure 4: The network dies early resulting in a large MAD of 0.355.

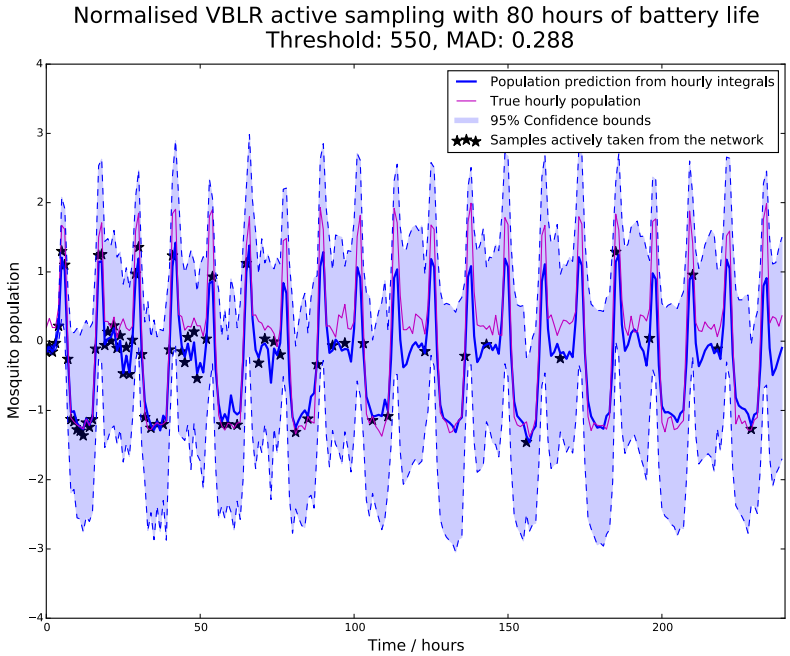


Figure 5: Network that lasts the entire data set and has a MAD of 0.288 in comparison to figure 4.

3.1 Measuring the performance of the network

It is a challenge to measure the success of the active sampling algorithm. This challenge arises because a network that accurately predicts the true mosquito population but dies early, must be compared to a network that inaccurately predicts the true distribution but survives for much longer. The accuracy of a network that expends all of its battery too quickly will perform better over the range it has sampled from, but will do worse over the rest of the data as its measurements will drop to zero with no battery life. This paradigm is shown in figures 4 and 5.

A measure that can be used to determine how well the network is doing is to take the median absolute deviation (MAD) between the true mosquito population, p_{true}, and the estimated population, p_{est}. The median absolute deviation is defined as:

$$\text{MAD} = \text{median}\left(|p_{true} - p_{est}|\right). \tag{7}$$

The MAD is taken over the entire range of the data, causing a network to be penalised if it dies early, even if it is accurate over the heavily sampled region.

It is also important to measure the performance of the active sampling by looking at the lifespan of the network. This can be formulated in terms of the percentage improvement on the nominal battery life of the network. Both the percentage improvement on the battery life and the MAD can be used to make a decision on the appropriate value to set for the sampling threshold. The results of applying these measures of utility are shown in the following section.

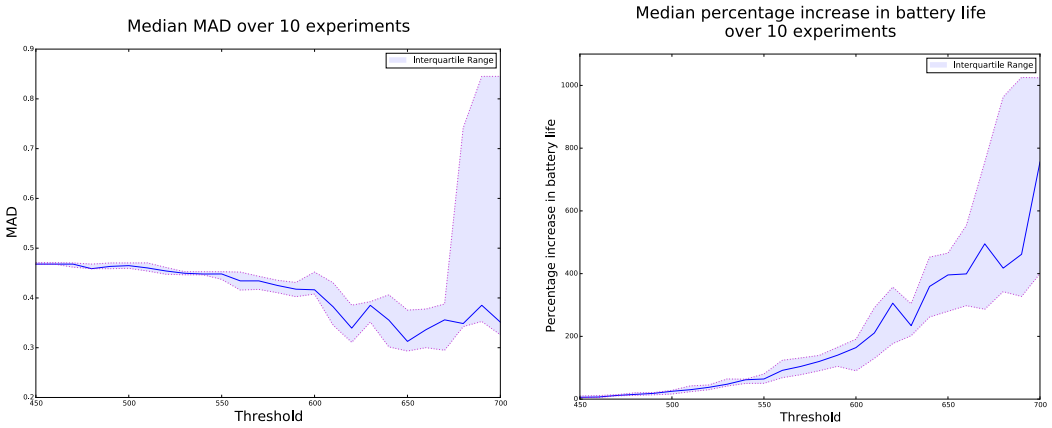
4 Results

Each experiment consists of running the model over a range of thresholds. These experiments are then repeated ten times using different random seeds. The results are then analysed by looking at the interquartile range and median over these experiments.

Following this methodology, figure 6a demonstrates how the MAD decreases with an increasing sampling threshold until a certain point. The greatly increased interquartile range after the threshold of 670 in figure 6a is significant in making a decision as to the best threshold to select for this network configuration. The steep rise of the upper quartile means that there is a greater likelihood larger errors occurring than at lower thresholds. Therefore indicating that the advantage in accuracy of selecting a larger threshold no longer applies past this point. Additionally, figure 6b shows the percentage increase on the nominal battery life with the increasing threshold. This is as expected because a larger threshold encourages fewer samples to be taken. The interquartile range grows as well, implying that we can be less certain of the expected lifespan of the network with larger sampling thresholds.

5 Conclusion

The results in the previous section demonstrate the compromise between increasing the lifetime of the network and the level of accepted error between the true and predicted mosquito distribution. Looking at figures 6a and 6b, a larger sampling threshold implies that the network lasts longer until a threshold of 650, as the median and the upper quartile of the MAD across the 10 experiments are both lower until a threshold of 650. It is also important to note that as the interquartile range grows with larger thresholds. The implication is that we are increasingly less certain of achieving better results in terms of



(a) (b)

Figure 6: The two plots show the results of running the model for a range of thresholds and a range of random seeds. The inittal battery life given to the network was 60 hours and the data set consisted of 674 hours of readings. The median and interquartile range are calculated over the ten results available at each threshold, where each result corresponds to a different random seed.

the MAD and the network's lifespan. Therefore it is recommended to choose a threshold that we expect will give a suitable level of accuracy and increased battery life along with a sufficiently low upper quartile. Continuing from our observation that the upper quartile increases after 650, this threshold could be a suitable choice for this particular network. The applied methodology generalises to other network configurations and can be used to improve their performance.

In this paper a basic sensor network to detect mosquitoes has been modelled and the constraint of improving the life of the network has been covered through active sampling. The results of this paper have been useful in exploring a way in which active sampling could be used in the HumBug project for coordinating the distributed devices. Modelling the sensor network has also highlighted many challenges that need to be overcome in future network design. The decision to set initial parameters, such as the number of sensors, the swarm locations and the sensor detection radii played a significant role in the network's behaviour. Better insight is needed for more accurate models in future work. However, as more mosquito data is collected, our knowledge of selecting these parameters will become clearer.

Another interesting area to be explored is the application of reinforcement learning to allow each sensor to learn when to expend and preserve battery life. Decentralising the control of the network may improve battery life as switching on the entire network to take a sample may not be the most efficient way to save battery whilst covering the largest possible area. Furthermore treating the sensors as agents and taking the game theoretic point of view could allow the sensors to start competing with each other to switch off, which could also improve the lifespan and coverage of the network.

Overall the technique of active sampling has increased the lifespan of the network and this success points to applying active sampling on real data in the near future when made available.

References

[1] Alessandro Farinelli, Alex Rogers, and Nick Jennings. Maximising sensor network efficiency through agent-based coordination of sense/sleep schedules. 2008.

[2] Alex Rogers, Esther David, and Nicholas R Jennings. Self-organized routing for wireless microsensor networks. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 35(3):349–359, 2005.

[3] Rajdeep K Dash, Nicholas R Jennings, and David C Parkes. Computational-mechanism design: A call to arms. *Intelligent Systems, IEEE*, 18(6):40–47, 2003.

[4] HumBug: Mosquito Detection and Habitat Mapping for Improved Malaria Vector Modelling. http://humbug.ac.uk. Accessed: 09-05-2016.

[5] Marianne E Sinka, Michael J Bangs, Sylvie Manguin, Theeraphap Chareonviriyaphap, Anand P Patil, William H Temperley, Peter W Gething, IR Elyazar, Caroline W Kabaria, Ralph E Harbach, et al. The dominant Anopheles vectors of human malaria in the Asia-Pacific region: occurrence data, distribution maps and bionomic précis. *Parasit Vectors*, 4(1):89, 2011.

[6] Yanping Chen, Adena Why, Gustavo Batista, Agenor Mafra-Neto, and Eamonn Keogh. Flying insect classification with inexpensive sensors. *Journal of insect behavior*, 27(5):657–677, 2014.

[7] GPy. GPy: A Gaussian process framework in python. http://github.com/SheffieldML/GPy, since 2012.

[8] S Roberts, A McQuillan, S Reece, and S Aigrain. Astrophysically robust systematics removal using variational inference: application to the first month of Kepler data. *Monthly Notices of the Royal Astronomical Society*, 435(4):3639–3653, 2013.

[9] Christopher M Bishop. Pattern Recognition. *Machine Learning*, 2006.

[10] Michael A Osborne, Stephen J Roberts, Alex Rogers, Sarvapali D Ramchurn, and Nicholas R Jennings. Towards real-time information processing of sensor network data using computationally efficient multi-output Gaussian processes. In *Proceedings of the 7th international conference on Information processing in sensor networks*, pages 109–120. IEEE Computer Society, 2008.