

Inferring proximity from Bluetooth Low Energy RSSI with Unscented Kalman Smoothers

Tom Lovett¹, Mark Briers^{1, *}, Marcos Charalambides¹, Radka Jersakova¹, James Lomax¹, and Chris Holmes^{1, 2}

¹The Alan Turing Institute, London, U.K.

²University of Oxford, Oxford, U.K.

*Corresponding author: Mark Briers, mbriers@turing.ac.uk

July 7, 2020

Abstract

The Covid-19 pandemic has resulted in a variety of approaches for managing infection outbreaks in international populations. One example is mobile phone applications, which attempt to alert infected individuals and their contacts by automatically inferring two key components of infection risk: the proximity to an individual who may be infected, and the duration of proximity. The former component, proximity, relies on Bluetooth Low Energy (BLE) Received Signal Strength Indicator (RSSI) as a distance sensor, and this has been shown to be problematic; not least because of unpredictable variations caused by different device types, device location on-body, device orientation, the local environment and the general noise associated with radio frequency propagation. In this paper, we present an approach that infers posterior probabilities over distance given *sequences* of Received Signal Strength Indicator (RSSI) values. Using a single-dimensional Unscented Kalman Smoother (UKS) for non-linear state space modelling, we outline several Gaussian process observation transforms, including: a generative model that directly captures sources of variation; and a discriminative model that learns a suitable observation function from training data using both distance and infection risk as optimisation objective functions. Our results show that good risk prediction can be achieved in $\mathcal{O}(n)$ time on real-world data sets, with the Unscented Kalman Smoother (UKS) outperforming more traditional classification methods learned from the same training data.

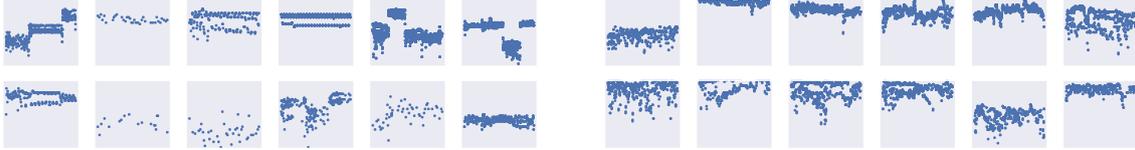
1 Introduction

There has been recent global interest in the use of Bluetooth Low Energy (BLE) Received Signal Strength Indicator (RSSI) as a proximity sensor. This is motivated by the international Covid-19 pandemic and the use of mobile phone applications to help control infection propagation through the population. At the time of writing, many of these applications are using Bluetooth Low Energy (BLE) to infer whether people are close together for prolonged periods of time, since proximate, prolonged exposure to an infected person correlates with the probability of infection [8, 27].

Unfortunately, BLE RSSI is a very noisy sensor of proximity. Due to the usual vagaries of radio frequency propagation in general environments, e.g. multipath, reflection, shadowing and fading, it becomes challenging to infer proximity from observed values without taking into account uncertainties in the data generating process.

To illustrate the highly variable behaviour of RSSI values, consider the plots in Figure 1. These all show RSSI values recorded at a fixed distance (1m for the Trinity College data sets, and 6ft for the MIT data sets) over time. Notice the extreme shifts and large variances; these are due to a multitude of sources, e.g. device type, orientation, position on body and the person’s local environment.

There have been various attempts to work with RSSI in a principled manner in Covid-19 mobile applications. In the popular Exposure Notification system [1, 3] used by Google and Apple devices, RSSI is “discretised” into buckets, and bucket thresholds are set by application developers. The documentation [2] reasons that RSSI is useful for inferring close proximity, but not at all effective for inferring larger distances.



(a) Trinity College Dublin data, from [16].

(b) MIT PACT data [15].

Figure 1: Samples of RSSI data over time from two sources. The Trinity College data here are all recorded at a proximity of 1m, and the MIT data at 6ft. All plots across both sources share the same y -axes $[-100; 50]$.

In this paper, we present a probabilistic model for inferring proximity using BLE RSSI observations. This model uses a Unscented Kalman Smoother (UKS) [11, 12], which takes advantage of sequential RSSI data, and models multiple sources of uncertainty in the data distribution variance. For the data distribution, we use a Gaussian process that maps distance to observations, and present two general forms: a generative model, which directly models the sources of shifts and variance in the observations; and a discriminative model, which learns a suitable observation model from training data. Setting this paper’s contribution in context, the estimation of a distribution over proximity is just one technical problem that needs to be addressed; the reader is also referred to [22] for a broader discussion around the ethical considerations around contact tracing app development.

2 Related work

Since the Covid-19 outbreak, there has been a surge of interest in proximity inference using BLE RSSI data. Since risk of infection is a function of, *iter alia*, proximity between individuals and the duration of contact [8, 4], automatically inferring these two properties for any contact “event” is crucial.

Perhaps the most notable work exploring the effects of BLE RSSI in real-world environments, and its potential as a sensor of proximity, are the series of papers by Trinity College Dublin. For example, in [16], the authors demonstrate the extreme shifts and variation in RSSI over a variety of contexts, including on public transport, in supermarkets, walking in public streets and sitting at desks. There are also scenarios demonstrating the shifts in RSSI from simply putting a mobile device in a pocket or a bag. The data in Figure 1 are taken from this paper.

In [17], the authors show that in this public transport environment with 60 device pairs, the Exposure Notification system with particular parameter settings did not detect genuine contact events, though minor improvements were made through parameter variation. Further evidence of the difficulties of using RSSI is illustrated in [18], with the best performing parameter settings achieving the equivalent of random selection.

Other recent studies into Bluetooth for proximity detection include: [19], where further tests of Bluetooth in indoor and outdoor environments, as well as device concealment variations, show the volatility of RSSI and the ambiguity at larger distances; and [21], where BLE is used for proximity detection in the workplace, albeit with exposed bracelets and additional environment sensors, and good binary classification of contact events is achieved through varying scan windows at a cost of power consumption (the drop in performance for low-power, lower frequency methods even in a “good coverage” environment is noted).

Methods that make use of RSSI sequences include [25], which uses particle filtering on Bluetooth RSSI to infer proximity in idealised environments. The findings are unlikely to translate to everyday mobile phone use however, since they were obtained from sensor networks with different Bluetooth hardware, where the setup was designed specifically for object tracking. Other methods have used Kalman filters, which assume linear transforms for state transition and observation. In [28], a Kalman filter was applied to the indoor location estimation problem but, again, the hardware and experiment setup used (high-power class 1 Bluetooth sensor networks set up for tracking) do not translate to real-world mobile device use. Similar studies used augmented sensors, e.g. an inertial sensor [30], in environments designed for tracking, e.g. [31] and some have considered particle filters with Gaussian processes for sequential modelling [10].

In direct response to the Covid-19 crisis, other approaches to proximity detection with realistic

context and BLE hardware have been studied. These include Gorce, Egan and Gribonval [9], who use calibrated BLE RSSI, where shifts due to device type, position and environment are considered, as well as probabilistic modelling of fading and shadowing. The authors then use Bayesian inference to compute a posterior distribution over distance given (averaged, calibrated) RSSI observations. This posterior uses a constrained uniform prior with Gaussian data distribution over the RSSI observations. Different estimators are derived from this posterior, such as the maximum *a posteriori* and maximum likelihood distance. These are then used in various risk scoring approaches, and experiments show reasonable risk inference using this method. This work is similar to ours in its Bayesian approach, though it does not take sequences into account. Moreover, we model shifts as random variables in the data distribution rather than through correction and averaging.

3 Posterior proximity inference

Given a sequence of observed RSSI random variables $R_1; \dots; R_{T^0}$, we wish to compute the probability distribution over device proximity, or distance, at each observation, i.e. $D_1; \dots; D_{T^0}$. Since RSSI data are likely to be aperiodic, bursty and unreliable, we might wish to treat $D_1; \dots; D_{T^0}$ as a subsequence of a larger, periodic sequence $D_1; \dots; D_T$, where $T^0 \ll T$ and also infer the proximity at points where observations are not present (see Section 3.1.1).

Since (negative) RSSI appears to have a longer tail to -7 in empirical data, e.g. Figure 1 and the plots in [9], and that BLE typically has transmission power -10 dBm, we model R as a log-normal random variable. This is in contrast to popular radio propagation models such as the log-distance path model, which assumes the variability in R follows a Gaussian distribution. For completeness, we have replicated all results in this paper under the assumption of R being a Gaussian random variable (see Appendix C), and the log-normal model appears to be more resilient to fluctuations in RSSI, as discussed in Section 8.

Under this log-normal model, we are interested in modelling the data distribution of the Gaussian random variable $X = \log(-R) \in \mathbb{R}$ conditioned on a distance (or proximity) variable $D \in (0; 1)$, where $R \in (-1; 0)$ is RSSI in dBm at distance d^l . We assume this conditional distribution is characterised by a collection of D -dependent parameters $\Theta(D)$,

$$F_{X|D}(X | D; \Theta(D));$$

and, given this data distribution for $X | D$ along with the sequence of observations $X_1; \dots; X_{T^0}$, our goal is to infer the posterior distribution over D_t at each time index $t \in 1; \dots; T^0$

$$F_{D_t|X_1; \dots; X_{T^0}}(D_t | X_1; \dots; X_{T^0}; \Theta(D_t));$$

We further assume that this distribution (and the data distribution) admits a density with respect to Lebesgue measure, and we compute

$$p_{D_t|X_1; \dots; X_{T^0}}(D_t | X_1; \dots; X_{T^0}; \Theta(D_t)) / p_{X_1; \dots; X_{T^0}|D_t}(X_1; \dots; X_{T^0} | D_t; \Theta(D_t)) p_{D_t}(D_t);$$

This is a classic problem in dynamical systems' theory, and is well suited to methods such as Kalman filtering, smoothing and other derivatives. The choice of method depends on the form of the data distribution, and performance depends heavily on the quality of the data distribution, i.e. how closely it matches nature's "true" distribution.

3.1 Unscented Kalman filtering and smoothing

The Kalman filter and smoother are classic methods for performing posterior inference over latent variables $\mathbf{x}_1; \dots; \mathbf{x}_T \in \mathbb{R}^m$ given discrete sequences of observed vectors in $\mathbf{z}_1; \dots; \mathbf{z}_T \in \mathbb{R}^n$. The traditional Kalman smoother assumes that the latent sequence has the Markov property, all transforms are linear and that all stochastic sources are Gaussian. The state transition model, assuming no control inputs, is

$$\mathbf{x}_{t+1} = \mathbf{A}_{t+1}\mathbf{x}_t + \mathbf{w}_{t+1}; \tag{1}$$

where $\mathbf{w}_t \sim N(\mathbf{0}; \mathbf{Q}_t)$. The observation model is

$$\mathbf{z}_t = \mathbf{B}_t\mathbf{x}_t + \mathbf{v}_t; \tag{2}$$

¹We assume BLE transmission power is always < 0 dBm

where $\mathbf{v}_t \sim N(\mathbf{0}; \mathbf{R}_t)$.

Unfortunately, in our application of inferring proximity from BLE RSSI, the observation model is non-linear and the latent variables D only have non-negative support. We can work around the latter problem by assuming $D_t \geq 0$ and transforming D_t to its absolute value, i.e. $|D_t|$ – this implicitly assumes that the transition distribution is *folded* normal, rather than normal. The nonlinearity of the observation model leads us to use an extension to the traditional Kalman smoother: the Unscented Kalman Smoother (UKS) [11, 29, 5]. The UKS uses deterministic inspired sampling to allow nonlinear transforms in the model. Our transition model for the UKS is

$$D_{t+1} = |D_t + w_{t+1}|; \quad (3)$$

where $w_t \sim N(0; q_t)$. This model is equivalent to assuming that the two devices are each performing an independent Gaussian random walk, and that the relative proximity transition follows a folded normal distribution. The key parameter in this model is q , the variance of the change in proximity between time steps. The observation model is

$$X_t = |D_t| + v_t; \quad (4)$$

where $v_t \sim N(0; r_t)$. In other words, $X_t \sim N(|D_t|; r_t)$.

3.1.1 Posterior imputation

One of the key advantages of the UKS, and dynamical systems in general, is the ability to infer the posterior distribution over D_t , even when an observation X_t does not exist. This is well suited to RSSI data, which are bursty, aperiodic and unreliable. This illustrates another benefit of posterior inference over sequences of observations, rather than single observations independently. Other approaches use averaging to smooth observations and inferred values over time windows, e.g. [9], but the sequential nature of the UKS allows for more principled imputation, where observations either side of a ‘‘gap’’ induce a more realistic trend in the inferred values.

3.2 Choosing a suitable data distribution

Assuming the data distribution

$$F_{X|D}(X | D; \Theta(D))$$

is Gaussian, i.e.

$$X | D \sim N(|D|; r(D; r)); \quad (5)$$

with $f; r \geq \Theta(D)$ then, for distances d ,

$$(X_d) | d \sim N(0; 1);$$

is a Gaussian process and, for any given d , the Gaussian for X_d depends entirely on the hyperparameters $\Theta(D)$. Thus, we can encode knowledge of the distribution of X at certain distances in our choices for the hyperparameters. If we have access to appropriate training data, we can use these data to learn $\Theta(D)$ for RSSI behaviour in general environments. There are two main approaches for doing this: a discriminative approach, which learns representative parameters from training data; and a generative approach, which models directly the sources of RSSI variability. We consider both approaches and assess their performance in subsequent sections.

4 Data distribution form

In this section, we outline a model for the Gaussian data distribution

$$F_{X|D}(X | D; \Theta(D)):$$

Unfortunately, the vagaries of radio frequency propagation within different environments make physical modelling of this distribution very difficult.

Empirical data, e.g. [6, 9, 15, 16] suggest that the distribution of R in a ‘‘clean’’ environment, e.g. an anechoic chamber, has a unimodal, asymmetric form, with a long tail towards ∞ . We assume that the distribution of X is unimodal and symmetric about the mode, and a member of the location-scale family of distributions.

For a given distance $D = d$, we assume the existence of a fixed function $f : (0; 1) \rightarrow \mathbb{R}$, which captures the physics of radio propagation in free space as a function of distance. We use a simple RSSI propagation model, which approximates line-of-sight received power in free space using the Friis transmission equation,

$$P_r = P_t G_t G_r \frac{\lambda^2}{4 d^2};$$

where P_r is received power (in W); P_t is transmitted power; G_t and G_r are transmitter and receiver gains respectively; d is distance between transmitter and receiver (in m); and λ is wavelength (in m). We use the decibel conversion

$$g(d) = 20 \log_{10} \frac{\lambda^2}{4 d^2}; \quad (6)$$

where $\lambda = 0.125$ is Bluetooth wavelength in metres². We assume the transmitted power to be 0dBm, and that antenna gains for the transmitter and receiver are captured in the shift variables below. The base function $f(d)$ is then

$$f(g(d)) = \log(g(d)); \quad (7)$$

In our model, this function can be shifted by a finite number N_s of independent random variables $Y_i | d$ – which may represent, e.g. antenna orientations, device model differences and changes in the physical environment – plus some zero-mean unattributable, independent, distance-invariant noise Z . With these forms and assumptions, we have, given $D = d$

$$X | d = f(d) + \sum_{i=1}^{N_s} Y_i | d + Z;$$

and,

$$\begin{aligned} \mathbb{E}[X | D = d] &= f(d) + \sum_{i=1}^{N_s} \mathbb{E}[Y_i | d] + \mathbb{E}[Z]; \\ &= f(d) + \sum_{i=1}^{N_s} \int_{\mathbb{R}} y_i F_{Y_i | d}(dy_i) + \int_{\mathbb{R}} z F_Z(dz); \end{aligned} \quad (8)$$

with

$$\begin{aligned} \text{Var}(X | D = d) &= \sum_{i=1}^{N_s} \text{Var}(Y_i | d) + \text{Var}(Z); \\ &= \sum_{i=1}^{N_s} \int_{\mathbb{R}} (y_i - \mathbb{E}[Y_i | d])^2 F_{Y_i | d}(dy_i) + \int_{\mathbb{R}} z^2 F_Z(dz); \end{aligned} \quad (9)$$

5 Generative model

In this section, we outline a generative model for certain shift variables Y_i and noise variable Z , each of which we assume to have Gaussian mixture form with K_i components at distance d ,

$$Y_i | d = \sum_{k=1}^{K_i} N_{k; \frac{2}{k}}; \quad (10)$$

and – in general – unknown d -specific K_i , k and $\frac{2}{k}$. For each variable Y_i , we assume we have access to some empirical observation data for each component k : D_k (which could be empty, i.e. $D_k = \emptyset$). We place a conjugate normal-inverse-gamma prior over each Gaussian component's parameters $k; \frac{2}{k}$ to obtain the posterior distribution,

$$k; \frac{2}{k} | D_k; k \sim \text{NIG}(m_k; k; k; k);$$

²This is for the 2402MHz advertising channel. Future work may wish to also consider the 2426MHz and 2480MHz channels, which equate to 0.123m and 0.121m wavelengths respectively.

and a conjugate Dirichlet prior over the mixture components, to obtain the posterior

$$p_{Y_i | \mathcal{D}}(Y_i | \mathcal{D}_1, \dots, \mathcal{D}_K; \Theta) \propto \prod_{k=1}^K \pi_k p_{Y_i | \mathcal{D}_k}(Y_i | \mathcal{D}_k; \Theta_k);$$

and marginalise over the unknown parameters in Equation 10 to obtain the posterior predictive distribution

$$p_{Y_i | \mathcal{D}}(Y_i | \mathcal{D}_1, \dots, \mathcal{D}_K; \Theta) = \int \prod_{k=1}^K \pi_k p_{Y_i | \mathcal{D}_k}(Y_i | \mathcal{D}_k; \Theta_k) d\Theta; \quad (11)$$

where each component has the form

$$\begin{aligned} p_{Y_i | \mathcal{D}_k}(Y_i | \mathcal{D}_k; \Theta_k) &= \int_0^1 \int_0^1 p(Y_i | m; s) p(m; s | \mathcal{D}_k; \Theta_k) ds dm; \\ &= t_{2 - \nu} \left(Y_i | m_k; \frac{(1 + \nu) \nu}{\nu} \right); \end{aligned} \quad (12)$$

i.e. each component is $t(\nu; \cdot)$: a non-standard Student's t -distribution with ν degrees of freedom. Thus our concrete distribution for $Y_i | \mathcal{D}$ is an average mixture of Student's t -distributions. Using standard results from conjugacy and $\mathcal{Y} := \{y_1, \dots, y_{N_k}\} \sim \mathcal{D}_k$, with $\bar{y} := \frac{1}{N_k} \sum_i y_i$,

$$\begin{aligned} \mu_k &= \frac{\nu_0 \mu_0 + \sum_i y_i}{\nu_0 + N_k}; \\ \nu_k &= \nu_0 + N_k; \\ \sigma_k^2 &= \frac{\nu_0 \sigma_0^2 + \sum_i (y_i - \mu_0)^2 + \frac{N_k \nu_0}{\nu_0 + N_k} (\bar{y} - \mu_0)^2}{\nu_k}; \end{aligned} \quad (13)$$

and, for the mixture components, given any multinomial observations of component frequencies $\mathbf{x}_1, \dots, \mathbf{x}_{N_\alpha}$ with $x_{i;1} \sim \mathcal{D}_1; \dots; x_{i;K} \sim \mathcal{D}_K$

$$p_{\mathbf{x}_i | \mathcal{D}}(\mathbf{x}_i | \mathcal{D}) = \prod_{k=1}^K \pi_k p_{\mathbf{x}_i | \mathcal{D}_k}(\mathbf{x}_i | \mathcal{D}_k); \quad (14)$$

5.1 Computing $\text{Var}(Z)$

For the unattributable noise Z , we assume zero-mean Gaussian with unknown variance $\frac{1}{2}$. With zero-mean and a single Gaussian component, the derivation of the previous section shows that

$$p_Z(Z | \mathcal{Z}) = t_2 \left(Z | 0; \frac{1}{2} \right);$$

so that

$$\text{Var}(Z | \mathcal{Z}) = \frac{1}{1}; \quad (15)$$

and we require $\nu > 1$.

5.2 Y variables

We focus on three classes of Y : device type shifts caused by differences in mobile device hardware; shifts caused by antenna gain variations; and context shifts caused by device usage, e.g. position, location and environment.

Device type: different device types affect shifts and variance of X [9], so this is one example of a shift variable Y . It also appears that this shift is different for different sender/receiver pairs. Given N device types, i.e. specific makes and models, we have $K = N^2$ sender-receiver pairs. Following Equation 11, our distribution for $Y_i | \mathcal{D}$ is

$$p_{Y_i | \mathcal{D}}(Y_i | \mathcal{D}_1, \dots, \mathcal{D}_K; \Theta) = \int \prod_{k=1}^K \pi_k p_{Y_i | \mathcal{D}_k}(Y_i | \mathcal{D}_k; \Theta_k) d\Theta; \quad (16)$$

Given choices for prior hyperparameters κ , we can collect data on sender/receiver device shifts by measuring them empirically, e.g. in an anechoic chamber, and updating the posterior hyperparameters. We can also use mobile device market share data or survey data to update the Dirichlet parameter in Equation 14.

Antenna gain: the Friis transmission equation in Equation 6 usually includes terms for losses in received power due to directivity of the transmitter and receiver. We encode these losses in a random shift variable Y , with a single component. Thus Equation 11 becomes

$$p_{Y|d}(Y|D; \kappa) = t_2 \quad Y | m; \frac{(1 + \kappa)}{2}^{\frac{1}{2}};$$

and empirical data on directivity shifts can be used to update the posterior parameters.

Device position, location and environment: other sources of variability for X include device (antenna) position, e.g. orientation; device location, e.g. in pocket; and environment, e.g. indoors. Since these are typically not independent, the shift Y depends on the joint distribution of the variables P (position), L (location) and E (environment). If we assume the position, location and orientation variables take values in finite sets, then our mixture model will have $K = N_P \cdot N_L \cdot N_E$ components and we have the form of the posterior predictive mixture of Student's t -distributions in Equation 11. Again, we can use empirical data for each component D_k to update the posterior hyperparameters.

6 Discriminative model

The purpose of a discriminative model is to learn suitable parameters $\Theta(D)$ from training data. This learning process is an optimisation problem, and we wish to find suitable parameters that maximise a particular objective function.

6.1 Model form and parameters

There are a range of parametric forms that we could use for the discriminative model of the data distribution in Equation 5. We consider two here: the first is a scaled and shifted base function $f(d)$ with \mathcal{d} -invariant scale and shift parameters μ_1 and μ_2 respectively, i.e.

$$f(d; \mu) = \mu_1 f(d) + \mu_2; \quad (17)$$

and the second disregards the base function f and assumes a logarithmic form with \mathcal{d} -invariant scale and intercept, i.e.

$$f(d; \mu) = \mu_1 \log(d) + \mu_2; \quad (18)$$

For both forms, we also have the observation variance σ_r and transition variance q for the UKS. The parameters for optimisation are therefore $\Theta(D) = \{ \mu_1; \mu_2; \sigma_r; q \}$.

6.2 Proximity optimisation

The first objective function is to minimise the expected average mean-squared error between true distances $D_1; \dots; D_T$ and the expected value of the posterior distribution returned by the UKS inference process. That is, given N training data sets $D_1; \dots; D_N$, where $-$ to account for missing observations $-$ D_n contains a periodic sequence of true distances $(d_1; \dots; d_{T_n})$ and a generally aperiodic, *subsequence* of observations $(x_1; \dots; x_{T_n^0})$ with $T_n^0 \leq T_n$. We wish to find

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}} \mathbb{E}_D \left[\frac{1}{T_n} \sum_{t=1}^{T_n} \sum_{j=1}^{T_n^0} \mathbb{E} [D_t | x_1; \dots; x_{T_n^0}; \Theta] - d_t \right]^2; \quad (19)$$

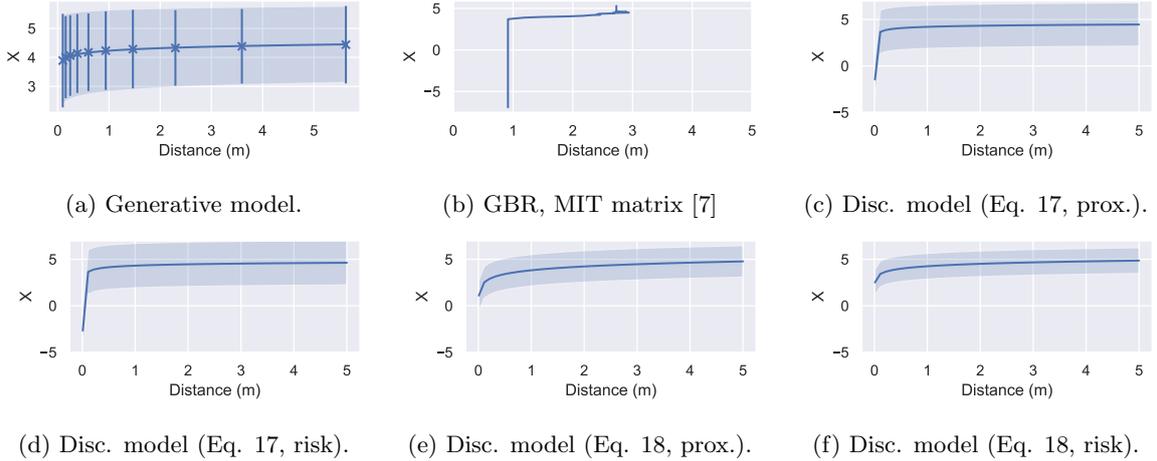


Figure 2: Gaussian process data distributions for the various models. The exception is (b), which shows the gradient boosted regressor prediction of distance from RSSI (note: the axes are reversed to align with the other plots, so d is a function of X here). The confidence intervals mark the 0.05 and 0.95 quantiles of the Gaussian distributions. The generative model shows X computed with $E[X | D]$ and $\text{Var}(X | D)$ at a finite set of d values. Interpolation is provided by standard Bayesian ridge regression on $\log(d)$. For the discriminative models, “prox.” means proximity optimised using Equation 19 and “risk” means risk optimised using Equation 21.

6.3 Risk error optimisation

The second objective function is to minimise the risk error. We use the risk score from [4] for one time step under the assumption of maximum infectiousness and minimum time decay as

$$(d_t) = \frac{\Delta t}{60} \min \left[1; \frac{1}{d_t^2} \right] ; \quad (20)$$

where Δt is the time step (in seconds) between periodic true distances, and search for parameters that minimise the expected average mean-squared risk error,

$$\hat{\Theta} = \underset{\Theta}{\text{argmin}} E_D \frac{1}{T_n} \sum_{t=1}^n \mathbb{X}_t^n (d_t) \quad E D_t | \mathbf{x}_1; \dots; \mathbf{x}_{T_n^0}; \Theta \quad \# \quad (21)$$

6.4 Optimisation approach

Unfortunately, the complexity of the UKS means that evaluating any objective function involves running a full smoothing process over each training data set D_n . For the n^{th} training set, this is $O(T_n)$ (since we have single dimensional latent and observation spaces) and so, for N training sets, a single evaluation of an objective function is $O(NT_{\max})$, where T_{\max} is $\max_n T_n$.

Because of this, we use Bayesian optimisation [26]. Bayesian optimisation uses a Gaussian process as a surrogate function to optimise low-dimensional objective functions with high evaluation cost. Since we have at most 4 model parameters to optimise, our problem is well-suited to the Bayesian optimisation approach. The experimental setup and results for the discriminative models are detailed in Section 7.5.

7 Model configurations and performance results

In this section, we apply both model types to various data sets. We first outline the configuration for each type and the data sets involved in parameter learning, before detailing the test data sets and presenting performance results on each for all models.

7.1 Gradient boosted regressor

As a benchmark for comparison, we trained a gradient boosted regressor on the MIT Matrix data set [7]. These are RSSI data captured in a variety of contexts at 8 different distances: 3;4;5;6;8;10;12 and

15ft. There are 107 files consisting of 118 individual pairwise interactions.

For training, we merged all RSSI points into one set, and used 3-fold cross validation with a test proportion of 0.33. For the gradient boosting, we used LightGBM [13] with Root Mean-Squared Error (RMSE) loss function on distance, 31 leaves and 100 iterations. The learned prediction function is shown in Figure 2 (b).

7.2 Discriminative model configuration

For discriminative model training data, we again used MIT’s matrix data set [7]. We use 3-fold, stratified cross-validation on the data *sets* to choose the optimisation parameters for each model. The stratification is set so that at least one data set from each recorded proximity appears in both the training and validation sets. For each data set, we resample to $\Delta t = 1$ s (we take the mean for multiple observations in a single time step).

For these results, we use 100 rounds of Bayesian optimisation over the full UKS from 10 initialisation points using a Matérn kernel for the Gaussian process with $\nu = 5/2$ and a small perturbation on the observed points (1×10^{-6}). For the model in Equation 17, we used the following search ranges: $\alpha_1 \in [0.8; 1.2]$; $\alpha_2 \in [0.5; 5]$; $r \in [0.3; 1.5]$ and $q \in [0.01; 0.05]$. For the model in Equation 18, we used: $\alpha_1 \in [0.01; 1]$; $\alpha_2 \in [3.5; 4.5]$; $r \in [0.2; 1.5]$ and $q \in [0.01; 0.05]$ For the optimisation process, we used the library in [23].

For the proximity optimisation objective function in Equation 19, we treat each data set with equal weight, so the expectation becomes the simple mean. For the risk optimisation function in Equation 21, we weight each dataset according to true risk, i.e. defining

$$w_n = \frac{\sum_{t=1}^T \mathbb{X}_n}{\sum_{t=1}^T (d_t)}$$

where \mathbb{X}_n is defined in Equation 20, we take the expectation in Equation 21 over D , with

$$p(D_n) / w_n$$

7.3 Generative model configuration

For the generative model, we compute $\mathbb{E}[\mathbb{X} | D]$ and $\text{Var}(\mathbb{X} | D)$ at a finite set of distances $\{d_1; \dots; d_K\} \in (0; 1)$. We fit posterior parameters $\Theta(d)$ where possible but, in the absence of empirical data, we use prior hyperparameters chosen to give appropriate uncertainty about the underlying generative processes.

7.3.1 Environment noise Z

For Z , we place a broad inverse-gamma prior over $\frac{2}{Z}$, with $\nu = 2$ and $\lambda = 1=10$. This reflects our uncertainty about RSSI in any arbitrary environment. The marginal variance of Z over all values for $\frac{2}{Z}$ (Equation 15) gives $\text{Var}(Z | \frac{2}{Z}) = 1=10$.

7.3.2 Device type shifts

We use the model in Equation 16, with \mathbb{X} set with counts of observations of UK mobile device market share data from 2019 [24], i.e. from the survey of $N = 2;123$ respondents,

$$k = N(N - 1)p_r p_t$$

where p_r is the proportion of the N respondents with device type r , and p_t the equivalent for type t , with $K = r^2 = t^2$. This hyperparameter controls our belief in the specific device types in a randomly selected pair for the UK.

GSMA have provided us with calibration offset data in dBm for 27 device makes and models. These are single observations of shifts at $d = 1$ m for certain transmitter/receiver device pairs in an anechoic chamber. We therefore do not have empirical data *sets* for our posterior predictive model in Equation 16, so we assume these figures set the α_0 hyperparameter for pair k , and that they were recorded with some uncertainty (encoded in the choices for $\alpha_0; \alpha_0; \alpha_0$). Since the figures are reported in dBm, we convert these to \mathbb{X} space as follows.

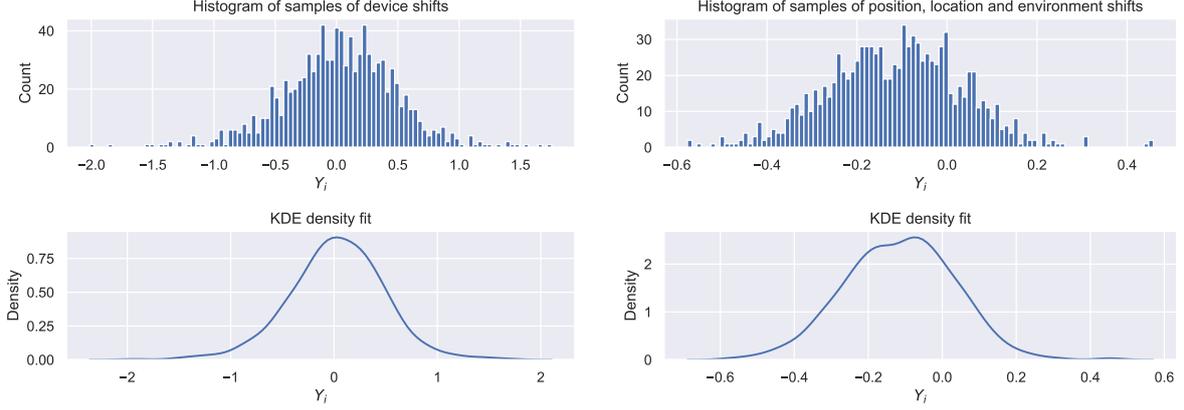


Figure 3: Left: 1:000 samples from $p_{Y_{ij}d}(Y_{ij} | \Theta)$ for device shifts at $d = 1\text{m}$ using Hamiltonian Monte Carlo (HMC) with No U-Turn Sampler (NUTS). Hyperparameters were set using anechoic chamber data for 729 (27^2) device pairs. σ_0 was set using UK mobile device market share data (see text). Right: 1:000 samples from $p_{Y_{ij}d}(Y_{ij} | D_1, \dots, D_K; \Theta)$ for assumed distance-invariant device position, location and environment shifts using HMC with NUTS. Hyperparameters were set using the MIT PACT data set [15]. σ_0 was set using survey data on mobile device usage (see text).

We view each supplied shift κ as a d -invariant shift in the negative Friis transmission equation $g(d)$. So, for $\kappa > g(d)$ and $g(d) < 0$, this produces a corresponding shift κ in $f(d)$ as follows

$$\begin{aligned}
 \kappa &= f(g(d) + \kappa) - f(g(d)); \\
 &= \log \left(\frac{g(d) + \kappa}{g(d)} \right); \\
 &= \log \left(1 + \frac{\kappa}{g(d)} \right); \tag{22}
 \end{aligned}$$

and we see that a constant κ results in a κ that varies with d through $g(d)$.

We can therefore define our device type shift variable $Y_{ij} \in \mathbb{R}$, and assume that the supplied κ is the σ_0 parameter in Equation 12. $\sigma_0 > 0$ and σ_0 encode our uncertainty about Y_{ij} , and we set them to $\sigma_0 = 1$; $\sigma_0 = 2$ and $\sigma_0 = 1=10$ to give small variance about σ_0 . Samples from $p_{Y_{ij}d}(Y_{ij} | \Theta(d))$ for $d = 1\text{m}$ are shown in Figure 3.

7.3.3 Antenna gain shifts

In addition to the calibration data supplied by GSMA, an estimated gain figure (in dBm) for the calibration reference device was also supplied. We follow the approach of the previous section to convert this to X space,

$$= \log \left(1 + \frac{\kappa}{g(d)} \right); \tag{23}$$

and set the d -specific hyperparameter $\sigma_0 = \kappa$. We again choose $\sigma_0 = 2$; $\sigma_0 = 1=10$ and $\sigma_0 = 1$.

7.3.4 Device position, location and environment shifts

We use the model in Equation 11, with the following hyperparameter settings. We assume the environment factor is split into two: indoors and outdoors. We set the indoors' probability to $p(E = \text{indoors}) = 0.869$ and the outdoors' probability to the complement, which are taken from the National Human Activity Survey (NHAPS) [14].

For device location, we use the data from [20] and assume a mobile device is not concealed for 8 hours (sleep) plus 3.25 hours in active use plus 8 hours not in use but nearby, e.g. working; leaving 4.75 hours with the device being concealed in a pocket or bag. Thus, we set the concealed probability to $p(L = \text{concealed}) = 4.75=24$, and the not concealed probability to the complement.

For device position, we assume equal belief to all orientation angles in a 2D plane. So, for any finite K -partitioning of $[0; 2\pi)$ into intervals I_1, \dots, I_K , we set $p(P \in I_k) = 1/K$.

With these, the k^{th} component of the Dirichlet hyperparameter becomes, for realisation $(p; l; e)_k$,

$$\kappa = N\rho(e)\rho(l)\rho(e);$$

where N is a pseudocount, which we set to 10.

For the remaining posterior predictive hyperparameters, we use the PACT datasets provided by MIT [15]. These contain measurements for two “real world” environment classes (1 outdoor and 3 indoor, with the 3 indoor sets merged) over two transmit location classes (3 concealed and 1 in hand, with the 3 concealed sets merged) over 8 angles of orientation. We use these data to set the parameters in Equation 13 – with $\alpha_0 = 0$; $\alpha_1 = \alpha_2 = 1=10$ and $\alpha_3 = 2$ – as follows.

We use the reference data sets recorded over location (concealed and in-hand) and positions (8 angles) in an anechoic chamber. We assume these are observations of noisy reference RSSI, and fit a normal distribution to $X := \log(R)$,

$$X_{l;p} \sim N(\hat{\mu}_{l;p}; \hat{\sigma}_{l;p}^2); \quad (24)$$

where the parameters are the sample mean and (unbiased) variance for each of the 16 reference sets. For the 32 other data sets, we observe shift variables as follows. For a given data set $D_{p;l;e} = \{x_1; \dots; x_{N_{p;l;e}}\}$, we draw

$$\hat{x}_1; \dots; \hat{x}_{N_{p;l;e}} \sim N(\hat{\mu}_{l;p}; \hat{\sigma}_{l;p}^2);$$

and use the observed shifts $y_i = x_i - \hat{x}_i$. These data are then used to compute the parameters for the t -distributions using Equation 13. See Figure 3 for samples from the full distribution of Y_i using these posterior parameters, and the mixing weights’ distribution parameters described above.

7.3.5 Approximating $E[X | D]$ and $\text{Var}(X | D)$

With the distributions of $Y_i | D_1; \dots; D_K; \Theta$ and the value of $\text{Var}(Z | z)$ set in the previous sections, we can approximate the expectation and variance of X with Equations 8 and 9. For this we use HMC with NUTS to estimate the expectations in Equations 8 and 9. The resulting Gaussian process under the computed estimates is shown in Figure 2 (a).

Figure 4 shows the UKS with generative observation model tracking proximity from noisy simulated data using RSSI generated from devices in the MIT H0H1 data set [6].

7.4 Test data sets

For performance evaluation, we use the MIT H0H1 data set [6] and the Trinity College Dublin data sets from [16].

7.4.1 MIT H0H1 data set

This data set consists of RSSI captures from 26 “high risk” scenarios (H1), and 19 “low risk” scenarios (H0). We define a scenario to be an interaction between a device pair, and some of the raw data files contain multiple device interactions. There are iPhone and Android devices present in the data sets.

In the H1 scenario, participants were asked to stay within 6ft of each other for 15 minutes. In the H0 scenario, they were asked to stay at least 10ft apart for 15 min. In each scenario, participants were instructed to interact with each other normally in multiple environments, including: outdoors, indoors and sat at a table. Participants were also allowed to use their mobile phones as normal throughout the study.

We do not know if participants genuinely strayed over the instructed boundaries, nor do we know if each raw data file was intended to capture a single interaction. We include all mobile phone interactions in all files regardless.

Since we have proximity and risk bounds only, we cannot measure exact inference, but we know that the models should infer proximity ≤ 6 ft for H1, and ≤ 10 ft for H0, and thus “high risk” and “low risk” respectively.

7.4.2 Trinity College data set

This data set consists of RSSI readings in a number of settings, some of which are laboratory settings and some real-world settings. We use the real-world, or scenario settings, which consist of 14 sets of RSSI data in environments such as supermarkets, desks, public transport and walking in public. An approximate ground truth proximity is labelled for each set, though we do not know if participants

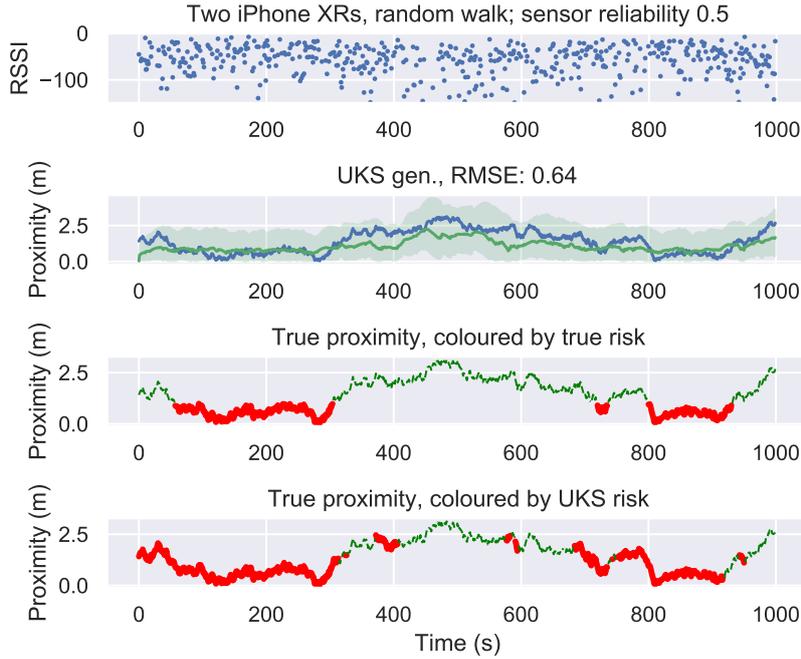


Figure 4: The UKS with generative observation model and $q = .09$ applied to simulated random walk data. Here, two iPhone XR devices undertake a random walk at on a circle with radius 2m for 1,000 seconds. We fit a sampling model with iPhone XR device types known (that is, without the mixture component over device type) from the MIT H0H1 data sets [6]. This results in a Gaussian process with $(\sigma) = 0.21 \log(\sigma) + 3.92$ and $\sigma^2 = 0.33$. RSSI samples are then generated at each time step from $\log\text{-normal}(\sigma, \sigma^2)$. In this example, half the observations are removed randomly to simulate imperfect sensor reliability. The topmost plot shows the RSSI data; the second plot shows the UKS with moment-matched gamma distribution 0.05 and 0.95 quantiles; the third and fourth plots show true and inferred risk coloration respectively – high risk, i.e. when within 1m of each other, is the thicker, solid red line; low risk is the thinner, dashed green line. Note the imputation of the UKS where there are missing observations.

rigidly adhered to this proximity throughout the capture. There are only Google Pixel 2 devices present in the data sets.

7.5 Performance results

Figure 5 shows the results for the MIT H0H1 data. Each point is a single scenario, coloured/marked by description. For the risk plots, we use relative risk, i.e. inferred risk minus the true risk bound. Since we do not know the true proximities – only their bounds – we can assess performance by visualising where the models place the scenarios above or below the 0 line.

For high risk, i.e. H1, points should be placed on or above the line, with increasing error the further below the line. For H0, points should be placed on or below the line, with increasing error the further above the line. A good proximity classifier would put all points above the black line for H1, H1 and all points below the line for H0, H0. A good risk classifier would put all points in H0, H1 below the line.

The Receiver Operating Characteristic (ROC) Area Under Curve (AUC) for the approaches are: gradient boosting regressor: 0.5; UKS g.: 0.823; UKS f.p.: 0.756; UKS f.r.: 0.6; UKS l.p.: 0.538; and UKS l.r.: 0.567.

For the proximity plot, a good proximity classifier would put all points in the H1 column below the black x-y line, and all the points in the H0 column above the black x-y line. A good risk classifier would put all points in the H0 column above the red dashed line (the H1 threshold).

Figure 6 shows example time series for the UKS with different models on an H1 and H0 scenario. Figure 7 shows inferred risk against true risk for the Trinity College Dublin scenarios in [16], the figures from which correspond to the plot legend labels in Figure 7.

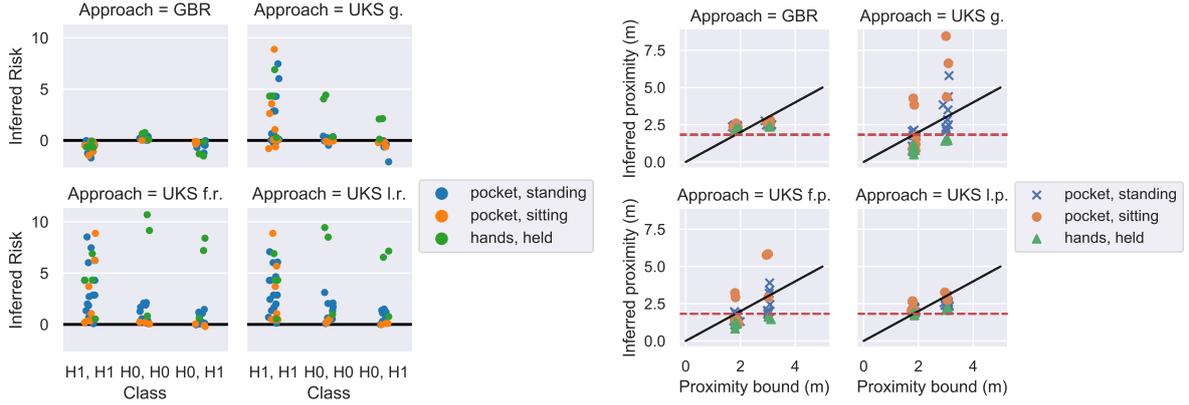


Figure 5: Left: relative risk, i.e. inferred risk minus true risk (bound), for MIT H0H1. H1, H1 means the high-risk scenario with high-risk threshold. H0, H0 is the low-risk scenario with low-risk threshold. H0, H1 is the low-risk scenario with high-risk threshold. Right: inferred proximity against true proximity (bound) for MIT H0H1. The two columns of points (with jitter) are the true bounds for H1 and H0 respectively. The red dashed line is the H1 proximity bound. GBR is the gradient boosted regressor; UKS g. is the generative model; UKS f.r./f.p. are the discriminative models in Equation 17 optimised for risk/proximity. UKS l.r./l.p. are the equivalent for Equation 18. See text for further details on plot interpretation.

8 Discussion

Here we discuss the implications and limitations of the results in the previous section. The key finding is that good prediction of proximity and risk can be achieved by treating RSSI sequences and using posterior inference of proximity D_t given the entire sequence of observations $x_1; \dots; x_{T^0}$ rather than x_t alone. By using a UKS, we can undertake this inference with nonlinear observation models; in this case Gaussian processes, which also encode uncertainty that propagates through to the posterior distribution over D_t . Given the single dimensions of both state space and observations, inference for a periodic sequence $D_1; \dots; D_T$ can be achieved in linear time, i.e. $O(T)$.

Using sequential modelling with Gaussian process data distributions outperforms simpler thresholding approaches such as the Exposure Notification API (cf. the effectively threshold-based gradient boosting AUC of 0.5 on MIT H0H1, which seems to corroborate the findings in [18]).

We next we compare the performance and suitability of proximity inference vs direct risk inference, before analysing the results of the generative and discriminative approaches. Next we acknowledge the implications of making a log-normal assumption for the distribution of \mathcal{R} , before discussing general limitations and potential areas for improvement in future work.

8.1 Proximity vs risk

Our chief intention is to infer posterior proximity given observed RSSI values, but there is an argument that predicting infection risk directly is more pertinent, especially given the application to the Covid-19 pandemic. The plots for the MIT H0H1 data in Figure 5 and the Trinity College data in Figure 7 show how the duration component of risk can make some encounters significantly more important to classify correctly. This implies that jointly inferring proximity and duration is arguably more important than proximity alone since, for example, a long duration at a farther proximity can equate to a shorter duration at closer proximity, and a classifier that seeks to predict close encounters well at the cost of incorrectly predicting farther ones may not achieve the desired effect of good overall infection risk prediction. We have not inferred duration here beyond the time duration of the scenarios in the test data, but an area for further work would be to better improve duration inference from real world RSSI observations.

8.2 Generative vs discriminative models

The results show that using the UKS with either a generative or discriminative model will likely outperform simple classification approaches, but there is a question as to which model is more appropriate.

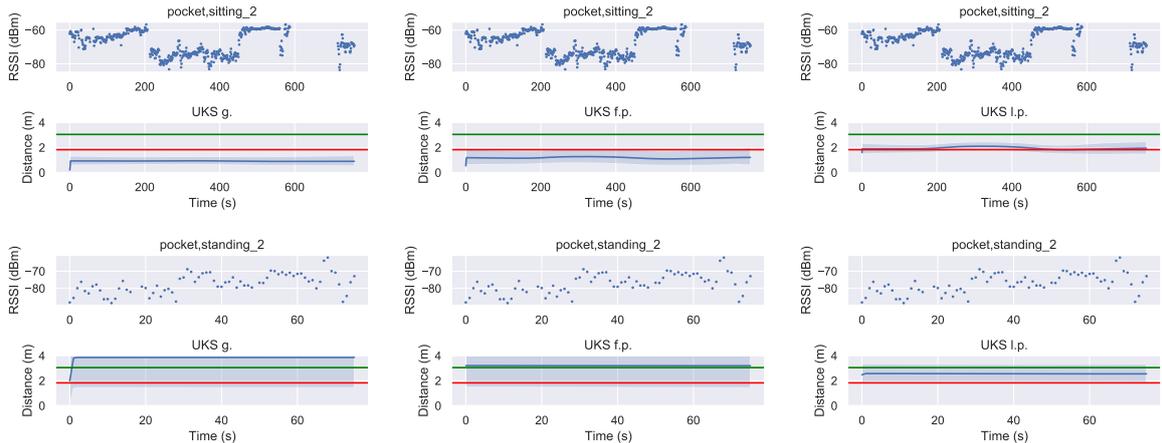


Figure 6: Time series of observed RSSI and UKS output (mean with 0.05 and 0.95 quantiles of a moment-matched gamma distribution) on one H1 example and one H0 example from MIT H0H1. Top row: H1 (high risk scenario); bottom row: H0 (low risk scenario); first column: UKS with generative model; second column: UKS with discriminative model (Equation 17); third column: UKS with discriminative model (Equation 18). The red horizontal line is the H1 threshold 6ft, and the green horizontal line is the H0 threshold.

The generative model is the best performing approach for the MIT H0H1 data (Figure 5), but the discriminative models outperform the generative model in the Trinity College data (Figure 7). There is an argument that the generative model is more general, since the discriminative models are limited by the training data, but the hyperparameters of the generative model are also computed from example data. There is arguably more flexibility to the generative model, since arbitrary numbers of shift variables Y_i can be added, but there is no strong evidence in our results to recommend choosing one over the other.

The question of which discriminative model to use is also not answered definitively, but the results in Figures 5 and 7 show marginally better performance using the form of Equation 17 over Equation 18. There are also limitations introduced by the search restrictions of Bayesian optimisation, and there may be better parameters for these models that were not found in the optimisation process.

It is perhaps unsurprising that the scenarios with the greatest error are where the mobile device is in the pocket with the individual sitting (Figure 5; false negatives), and where the device is in the hand (Figure 5; false positives).

8.3 Log-normal R vs Gaussian R

The main results in the paper assume that R is a log-normal random variable, but there is a valid argument that R should be normally distributed, e.g. in the log-distance path loss model for radio propagation. Our justification for using the log-normal distribution was based on empirical evidence of a long tail in observed real-world RSSI values, plus the assumption that transmission power should be at most 0dBm for BLE in mobile devices.

We have replicated all results using a normally distributed R , i.e. $X := R$, and these can be seen in Appendix C. There is a small difference in performance, with the log-normal model performing slightly better in general on the test data sets. The notable exception is the generative model on the Trinity College data (Figure 7 vs Figure 13), but the average performance of the log-normal appears to be slightly better. We conjecture that this is due to the resilience against RSSI fluctuations due to the long tail of the log-normal data distribution (compare the steadiness of the inferred proximity in Figures 6 and 12).

8.4 Limitations and potential improvements

Since we have not attempted to infer duration here, an obvious next step would be to focus on this; perhaps by attempting to partition RSSI data into sessions. We have also not considered other machine learning classifiers beyond a gradient boosting regressor, and it is entirely possible that a well-trained neural network could perform well, though we argue that much of the performance stems from the

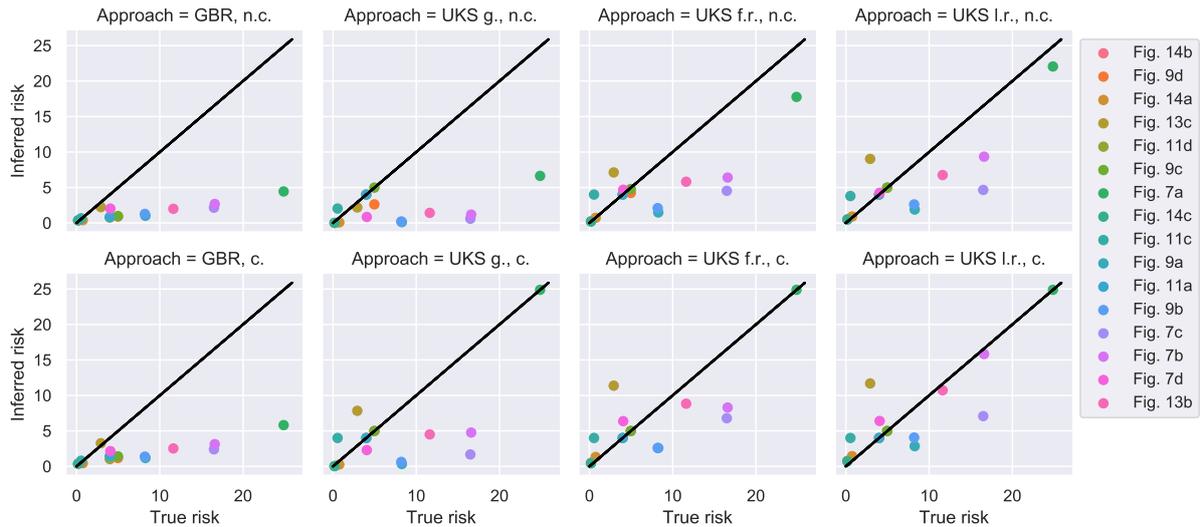


Figure 7: Trinity College data, with inferred risk against true risk. The top row shows the results for raw RSSI data. The bottom row shows the results when the RSSI are corrected with the knowledge of mobile device types (Google Pixel 2). GBR is the gradient boosted regressor; UKS g. is the generative model; UKS f.r./f.p. are the discriminative models in Equation 17 optimised for risk/proximity. UKS l.r./l.p. are the equivalent for Equation 18. The plot legend refers to figures in [16]. The n.c. and c. refer to “not corrected” and “corrected” respectively.

sequential modelling, and a sequential neural network may be a better choice. (These approaches are of course limited by access to good quality training data.) The other advantage of the UKS is uncertainty quantification, since we have posterior probability distributions over D_t and can report our confidence in the inference given the many sources of uncertainty in the data and underlying dynamics.

Other potential improvements could include: acquiring more, high quality training data for the models; exploring more complex UKS approaches, though it may be prudent to keep state space dimensions low since inference is naïvely $O(Td^B)$; optimising discriminative model parameters using approaches other than Bayesian optimisation; exploring data distribution forms other than log-normal (and Gaussian), e.g. the compound k distribution; and analysing performance as RSSI quality deteriorates, either due to noise or by intention for conservation of power.

The inference of proximity from BLE RSSI is a difficult problem, and more learning and validation data sets captured in varied scenarios (including from simulations) can only benefit any modelling approach.

9 Conclusion

In this paper, we presented a novel approach to inferring proximity from BLE RSSI using a UKS with Gaussian process data distribution. This is especially relevant to mobile phone applications designed to tackle the Covid-19 pandemic, which rely on good inference of infection risk; itself a function of proximity. We outlined two approaches to characterising the data distribution: a generative model, which directly computes sources of variability in observations; and a discriminative model, which optimises model parameters on example training data. There is no strong evidence to choose the generative approach over the discriminative one (or *vice versa*). Risk and proximity inference performance on two real-world data sets – MIT H0H1 and Trinity College Dublin – show that the UKS outperforms a more traditional gradient boosted regressor model. Our work to date offers an insight into well established mechanisms for probabilistic modelling of one of the key latent factors, that of proximity. We recognise that this needs to be considered in the wider context of health policy, ethical and other technical considerations, when responsibly deploying novel technology of this kind.

References

- [1] Apple API for Exposure Notification. <https://developer.apple.com/documentation/>

- exposurenotification. Accessed: 2020-07-01.
- [2] BLE Exposure Notifications Attenuations. <https://developers.google.com/android/exposure-notifications/ble-attenuation-overview>. Accessed: 2020-07-01.
 - [3] Google API for Exposure Notification. <https://developers.google.com/android/exposure-notifications/exposure-notifications-api>. Accessed: 2020-07-01.
 - [4] M. Briers, M. Charalambides, and C. Holmes. Risk scoring calculation for the current NHSx contact tracing app, 2020.
 - [5] M. Briers, A. Doucet, and S. Maskell. Smoothing algorithms for state–space models. *Annals of the Institute of Statistical Mathematics*, 62(1):61, 2010.
 - [6] C. Corey. MIT H0H1. <https://github.com/mitll/H0H1>. Accessed: 2020-07-01.
 - [7] C. Corey. MIT Matrix Data. <https://github.com/mitll/MIT-Matrix-Data>. Accessed: 2020-07-01.
 - [8] L. Ferretti, C. Wymant, M. Kendall, L. Zhao, A. Nurtay, L. Abeler-Dörner, M. Parker, D. Bonsall, and C. Fraser. Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing. *Science*, 368(6491), 2020.
 - [9] J.-M. Gorce, M. Egan, and R. Gribonval. An efficient algorithm to estimate Covid-19 infectiousness risk from BLE-RSSI measurements. Research Report RR-9345, Inria Grenoble Rhône-Alpes, May 2020.
 - [10] M. G. Jadidi, M. Patel, and J. V. Miro. Gaussian processes online observation classification for RSSI-based low-cost indoor positioning systems. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6269–6275. IEEE, 2017.
 - [11] S. J. Julier and J. K. Uhlmann. New extension of the Kalman filter to nonlinear systems. In *Signal Processing, Sensor Fusion, and Target Recognition VI*, volume 3068, pages 182–193. International Society for Optics and Photonics, 1997.
 - [12] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of 1995 American Control Conference-ACC'95*, volume 3, pages 1628–1632. IEEE, 1995.
 - [13] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3146–3154. Curran Associates, Inc., 2017.
 - [14] N. E. Klepeis, W. C. Nelson, W. R. Ott, J. P. Robinson, A. M. Tsang, P. Switzer, J. V. Behar, S. C. Hern, and W. H. Engelmann. The National Human Activity Pattern Survey (NHAPS): a resource for assessing exposure to environmental pollutants. *Journal of Exposure Science & Environmental Epidemiology*, 11(3):231–252, 2001.
 - [15] M. Krangle. BLE RSSI Various Static Configurations. <https://github.com/mitll/BLE-RSSI-Vari-ous-Static-Configurations>. Accessed: 2020-07-01.
 - [16] D. J. Leith and S. Farrell. Coronavirus Contact Tracing: Evaluating The Potential Of Using Bluetooth Received Signal Strength For Proximity Detection. 2020.
 - [17] D. J. Leith and S. Farrell. Measurement-Based Evaluation Of Google/Apple Exposure Notification API For Proximity Detection in a Commuter Bus. *arXiv preprint arXiv:2006.08543*, 2020.
 - [18] D. J. Leith and S. Farrell. Measurement-Based Evaluation Of Google/Apple Exposure Notification API For Proximity Detection In A Light-Rail Tram. SCSS Tech Report 26th June 2020, 2020.
 - [19] S. Liu, Y. Jiang, and A. Striegel. Face-to-face proximity estimation using Bluetooth on smartphones. *IEEE Transactions on Mobile Computing*, 13(4):811–823, 2013.

- [20] J. MacKay. Screen Time Stats 2019. <https://blog.rescuetime.com/screen-time-stats-2018/>. Accessed: 2020-07-01.
- [21] A. Montanari, S. Nawaz, C. Mascolo, and K. Sailer. A study of Bluetooth Low Energy performance for human proximity detection in the workplace. In *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 90–99. IEEE, 2017.
- [22] J. Morley, J. Cows, M. Taddeo, and L. Floridi. Ethical guidelines for covid-19 tracing apps, 2020.
- [23] F. Nogueira. Bayesian Optimization: Open source constrained global optimization tool for Python. <https://github.com/fmfn/BayesianOptimization>, 2014–.
- [24] S. O’Dea. Market share of smartphone manufacturers in the UK, 2019. <https://www.statista.com/statistics/387227/market-share-of-smartphone-manufacturers-in-the-uk/>. Accessed: 2020-07-01.
- [25] J. Rodas, C. J. Escudero, and D. I. Iglesia. Bayesian filtering for a Bluetooth positioning system. In *2008 IEEE International Symposium on Wireless Communication Systems*, pages 618–622. IEEE, 2008.
- [26] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pages 2951–2959, 2012.
- [27] C. Sohrabi, Z. Alsafi, N. O’Neill, M. Khan, A. Kerwan, A. Al-Jabir, C. Iosifidis, and R. Agha. World Health Organization declares global emergency: A review of the 2019 novel coronavirus (COVID-19). *International Journal of Surgery*, 2020.
- [28] F. Subhan, H. Hasbullah, and K. Ashraf. Kalman filter-based hybrid indoor position estimation technique in Bluetooth networks. *International Journal of Navigation and Observation*, 2013, 2013.
- [29] E. A. Wan and R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, pages 153–158. IEEE, 2000.
- [30] P. K. Yoon, S. Zihajehzadeh, B. Kang, and E. J. Park. Adaptive Kalman filter for indoor localization using Bluetooth Low Energy and inertial measurement unit. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 825–828, 2015.
- [31] C. Zhou, J. Yuan, H. Liu, and J. Qiu. Bluetooth indoor positioning based on RSSI and Kalman filter. *Wireless Personal Communications*, 96(3):4115–4130, 2017.

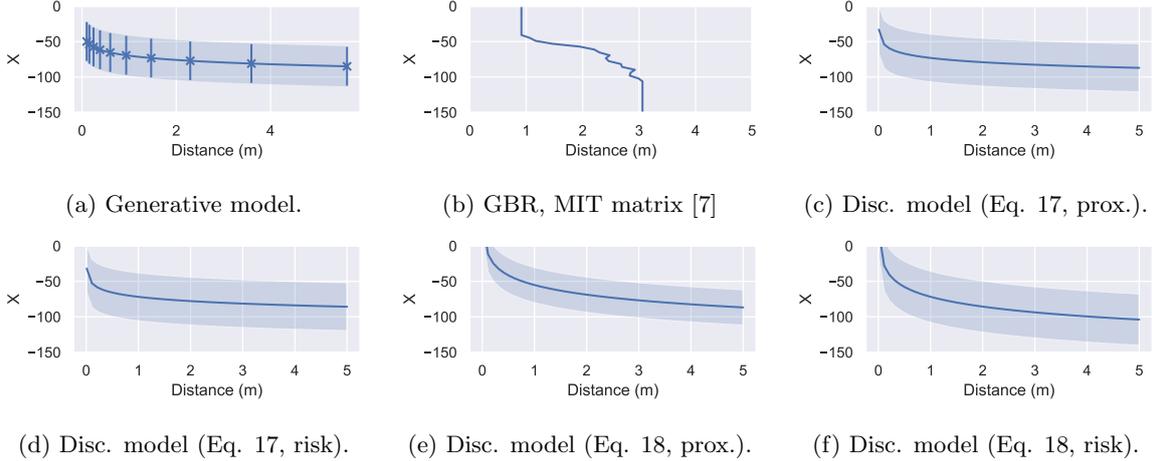


Figure 8: Gaussian process data distributions for the various models. The exception is (b), which shows the gradient boosted regressor prediction of distance from RSSI (note: the axes are reversed to align with the other plots, so d is a function of X here). The confidence intervals mark the 0.05 and 0.95 quantiles of the Gaussian distributions. The generative model shows X computed with $E[X | D]$ and $\text{Var}(X | D)$ at a finite set of d values. Interpolation is provided by standard Bayesian ridge regression on $\log(d)$. For the discriminative models, “prox.” means proximity optimised using Equation 19 and “risk” means risk optimised using Equation 21.

A Results using a Gaussian distribution on R directly

There is some debate about a suitable distribution for X . In the paper, we assumed a log-normal distribution on raw RSSI R and used a log transform to define the normally distributed $X := \log(R)$. This was motivated by asymmetric forms observed in empirical data, with long tails going to $-\infty$ and $-\infty$ under the assumption of at most a 0dBm transmission power – support on $(-\infty; 0]$.

In this supplement, we replicate all the results in the paper under a direct Gaussian model on R , i.e., we define $X := R$. This shows that the log-normal model is more robust to noise, but that performance on the test data sets is comparable.

B Data distribution form

The base function f in Equation 7 is now equivalent to Equation 6, i.e.

$$f(d) = g(d):$$

C Model configurations and performance results

In this section, we present the model configurations and results under the direct Gaussian observation model.

C.1 Discriminative model configuration

For these results, we use 100 rounds of Bayesian optimisation over the full UKS from 10 initialisation points using a Matérn kernel for the Gaussian process with $\nu = 5/2$ and a small perturbation on the observed points ($1 \cdot 10^{-6}$). For the model in Equation 17, we used the following search ranges: $\alpha_1 \in [1; 1.5]$; $\alpha_2 \in [100; 10]$; $r \in [0; 300]$ and $q \in [0.01; 0.05]$. For the model in Equation 18, we used: $\alpha_1 \in [20; 1]$; $\alpha_2 \in [100; 10]$; $r \in [0; 300]$ and $q \in [0.01; 0.05]$. For the optimisation process, we used the library in [23].

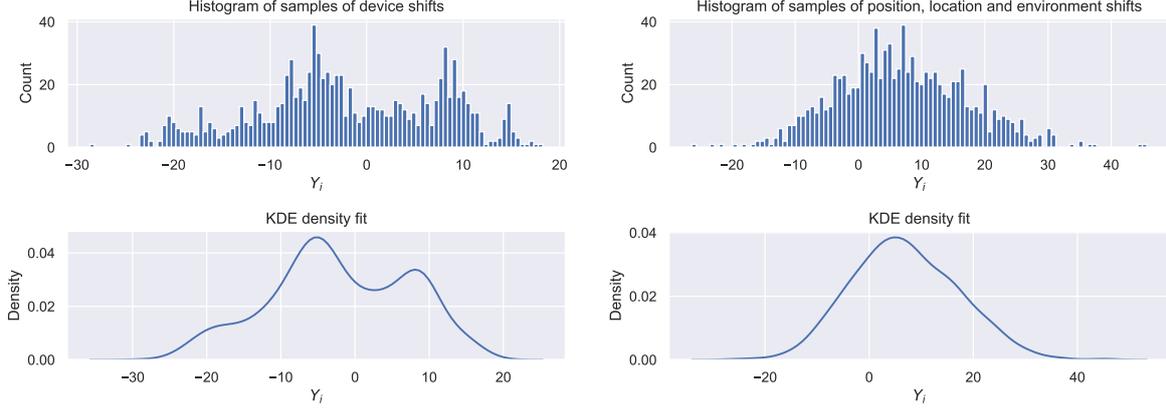


Figure 9: Left: 1;000 samples from $p_{Y_i|j,d}(Y_i|j,\Theta)$ for device shifts at $d = 1\text{m}$ using HMC with NUTS. Hyperparameters were set using anechoic chamber data for 729 (27^2) device pairs. was set using UK mobile device market share data (see text). Right: 1;000 samples from $p_{Y_i|j,d}(Y_i|j,D_1;\dots;D_K;\Theta)$ for assumed distance-invariant device position, location and environment shifts using HMC with NUTS. Hyperparameters were set using the MIT PACT data set [15]. was set using survey data on mobile device usage (see text).

C.2 Generative model configuration

The k shifts in Equations 22 and 23 simply map directly to negative k , i.e.

$$k = -k.$$

The normal distribution in Equation 24 is fit to $X := R$ rather than $X := \log(R)$. The variance in Equation 15 is set to 10dB, i.e. $\sigma^2 = 10$; $\sigma = 2$.

C.3 Performance results

The ROC AUC for the approaches are: gradient boosting regressor: 0.5; UKS g.: 0.728; UKS f.p.: 0.662; UKS f.r.: 0.728; UKS l.p.: 0.769; and UKS l.r.: 0.6. Figures 10-13 show the Figures from the main paper when using the Gaussian model.

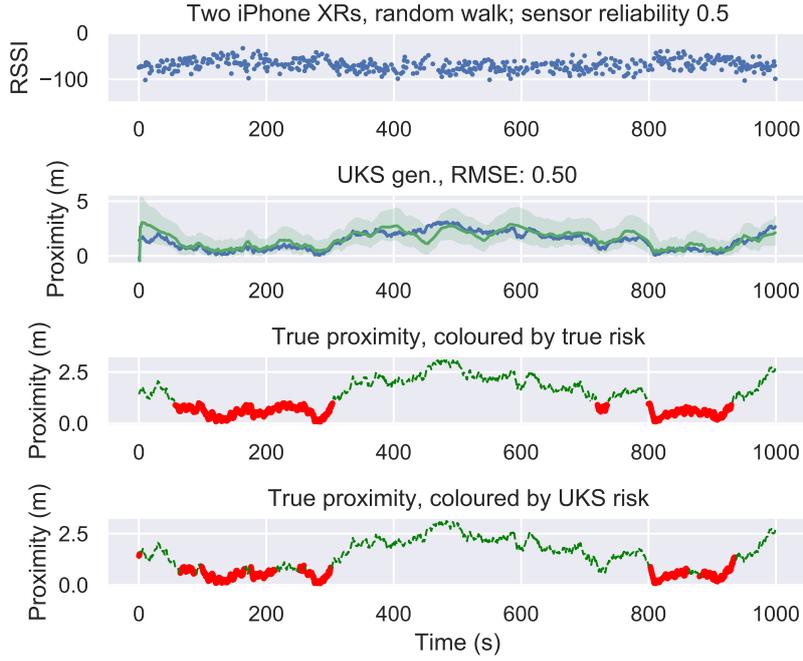


Figure 10: The UKS with generative observation model and $q = .09$ applied to simulated random walk data. Here, two iPhone XR devices undertake a random walk on a circle with radius 2m for 1,000 seconds. We fit a sampling model with iPhone XR device types known (that is, without the mixture component over device type) from the MIT H0H1 data sets [6]. This results in a Gaussian process with $(d) = 8.69 \log(d) - 69.9$ and $\sigma^2 = 184.91$. RSSI samples are then generated at each time step from $N(\mu(d); \sigma^2)$. In this example, half the observations are removed randomly to simulate imperfect sensor reliability. The topmost plot shows the RSSI data; the second plot shows the UKS with moment-matched gamma distribution 0.05 and 0.95 quantiles; the third and fourth plots show true and inferred risk respectively – high risk, i.e. when within 1m of each other, is the thicker, solid red line; low risk is the thinner, dashed green line. Note the imputation of the UKS where there are missing observations.

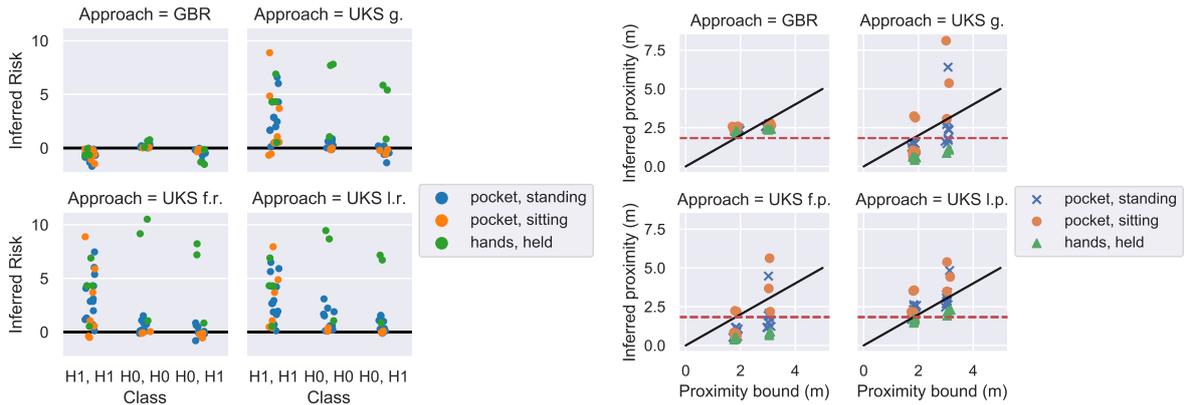


Figure 11: Left: relative risk, i.e. inferred risk minus true risk (bound), for MIT H0H1. H1, H1 means the high-risk scenario with high-risk threshold. H0, H0 is the low-risk scenario with low-risk threshold. H0, H1 is the low-risk scenario with high-risk threshold. Right: inferred proximity against true proximity (bound) for MIT H0H1. The two columns of points (with jitter) are the true bounds for H1 and H0 respectively. The red dashed line is the H1 proximity bound. GBR is the gradient boosted regressor; UKS g. is the generative model; UKS f.r./f.p. are the discriminative models in Equation 17 optimised for risk/proximity. UKS l.r./l.p. are the equivalent for Equation 18. See text for further details on plot interpretation.

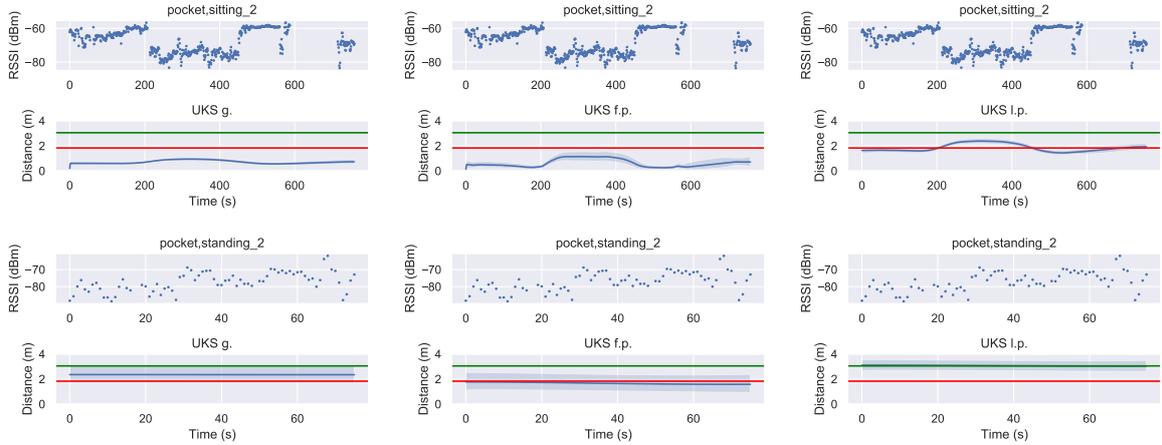


Figure 12: Time series of observed RSSI and UKS output (mean with 0.05 and 0.95 quantiles of a moment-matched gamma distribution) on one H1 example and one H0 example from MIT H0H1. Top row: H1 (high risk scenario); bottom row: H0 (low risk scenario); first column: UKS with generative model; second column: UKS with discriminative model (Equation 17); third column: UKS with discriminative model (Equation 18). The red horizontal line is the H1 threshold 6ft, and the green horizontal line is the H0 threshold.

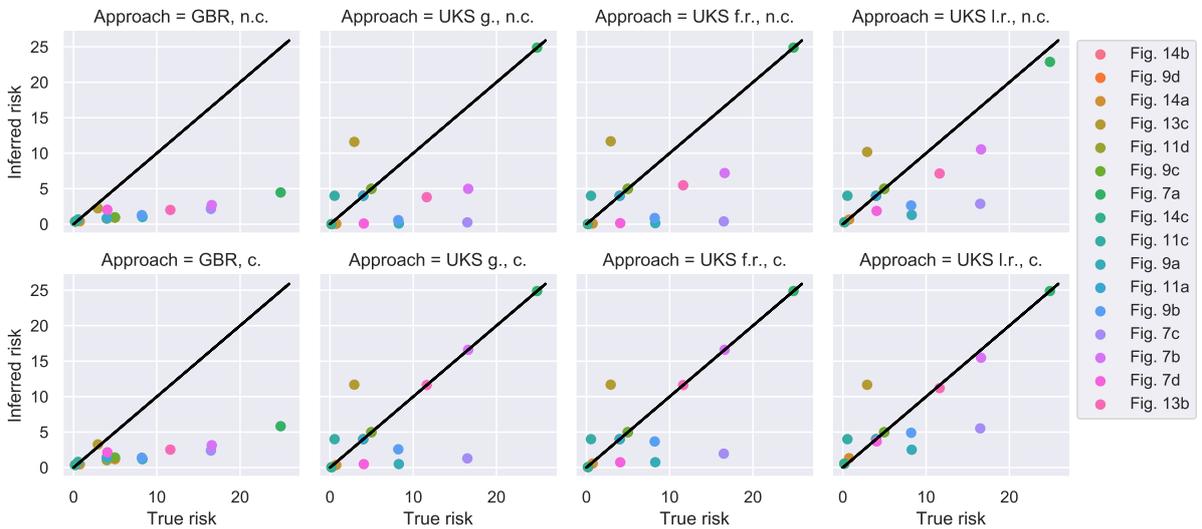


Figure 13: Trinity College data, with inferred risk against true risk. The top row shows the results for raw RSSI data. The bottom row shows the results when the RSSI are corrected with the knowledge of mobile device types (Google Pixel 2). GBR is the gradient boosted regressor; UKS g. is the generative model; UKS f.r./f.p. are the discriminative models in Equation 17 optimised for risk/proximity. UKS l.r./l.p. are the equivalent for Equation 18. The plot legend refers to figures in [16]. The n.c. and c. refer to “not corrected” and “corrected” respectively.