# A Bayesian non Parametric Approach to Time Series Modelling

Harry Petch-Smith[1,*]

[1]University of Birmingham, Department of Computational Biology, Birmingham, B15 2SQ, United Kingdom

## ABSTRACT

Time series modelling refers to a broad field in which the central aim is the analysis of data that varies as a function of time. A key example of an application of time series forecasting is in the prediction of blood glucose levels in diabetes patients. Prediction of blood glucose levels allows for the anticipation of detrimental events such as hypoglycemia and would therefor reduce patient risk. Whilst technology that monitors blood glucose levels has improved over time, the development of effective prediction algorithms is an unresolved issue. Complex machine learning methods such as deep learning neural networks provide high accuracy predictions but are complex and limited by hardware. In this study a simpler model is tested; the multiple output Gaussian process; which is capable of taking information from consecutive days to make more accurate predictions. This is done by calculating a covariance matrix using a kernel function that models the relationship between points within and between the data-sets. The model was tested on simulated data and continuous monitoring blood glucose data; accuracy was compared with a simpler single output Gaussian process. Overall the multiple output model performed better on the simulated data, however it performed poorly on the real world blood glucose data. This was likely due to the kernel function not encompassing the relationships between all data-points. More work needs to be done in the development of better kernel functions that handle multiple outputs.

## Introduction

In the past few decades advances in medicine have led to a dramatic increase in the quality of life of insulin dependant diabetes patients. Typically, insulin dependant patients lack insulin production or response leading to poor uptake of glucose into cells. Advances in insulin production have made it an abundant resource that many patients have access to, allowing them to compensate for the insulin they lose due to their diabetes. However, on average patients still experience problems, usually centred around meal intake and manual glucose monitoring[1]. Currently, it is mostly up to patients to monitor their glucose levels and administer insulin, which can lead to oversight and human error. This in turn can cause adverse events such as hypoglycemia (low glucose concentration in blood) that can in turn lead to coma or potentially death. There is more work that needs to be done to improve the lives of patients, especially since the prevalence of type 1 diabetes is increasing globally[2].

The next logical step in diabetes medicine is an artificial pancreas; a device that can continuously monitor blood glucose levels and administer insulin when required. This would dramatically reduce the number of adverse events experienced by the patient and allow them to live a normal life. The advent of continuous glucose monitoring (CGM) technology has made this much more feasible; patients blood glucose can be monitored in real time at regular intervals. Another key component of an artificial pancreas is the control algorithm which determines when insulin is administered. This works in tandem with CGM; reading the data and making decisions in real time. There are problems to overcome in regards to control algorithms. Whilst CGM technology has improved, there is often still lag between when a reading is reported and when the event that triggers it occurs. One approach to overcome this is known as the model predictive approach[3]. This involves making predictions about future blood glucose levels given current data, and would therefor artificially remove the effects of lag. However this approach needs to be accurate with

little room for error.

The model predictive approach falls under the broader category of time series forecasting; a method which aims to predict new values as a function of time. In the blood glucose example, we would assume that blood glucose values are mathematically related to the time-point at which they occur, and this relationship can be modelled. This task is performed computationally using a range of machine learning techniques such as auto-regression. The idea of applying these techniques to blood glucose monitoring has increased in popularity recently, with a range of CGM data-sets being made publicly available. One such data-set is Ohio TDM[4],on which a range of work has been performed using common time series forecasting models such as auto-regression and more complex deep learning neural networks[5].

A unique question in the processing of this style of data is how to process multiple outputs, especially if they are somewhat dependant on one another. For example, given the continuous glucose monitoring across two subsequent days, it would be beneficial to model the relationships within and between each day, to gain more accurate predictions. Patients may show an overall trend of eating meals at certain times, exercising at certain times etc, so taking these trends into account across multiple days should improve accuracy. Neural networks are notably good at handling multiple outputs, especially in relation to time series data, for example in air quality forecasting[6]. In 2019 a neural network was developed specifically for the purpose of blood glucose prediction, known as GluNet[7]. The model was trained on CGM data and produced some of the most accurate predictions that have been seen within the field; outperforming many more traditional methods. However, there are disadvantages to using such complex models; they are often computationally heavy and therefore have hardware limitations. They can also be considered black box models, since the process of prediction can be difficult to follow.[8]. A simpler model for processing multiple outputs is therefor required.

One such model is the Gaussian Process (GP). The GP is a non parametric method that estimates a set

of possible underlying functions that fit the data. Essentially, this forms a probability distribution across functions with a mean and variance/confidence interval. The mean function represents the most probable function that fits the data, and from this we can make predictions[9]. This is a Baysian method, where we make a prior assumption as to the multivariate Gaussian distribution of the data and update this based on observed data. The Gaussian process is a more robust method than other parametric forms of regression, as it is not limited by a single function e.g linear or logistic. It also gives confidence intervals to predictions which show a direct representation of how confident we can be with a model. Gaussian processes have a strong history of dealing with time series data[10].

Generally a Gaussian process is described by its mean $\mu$ and covariance matrix $\sum$. The covariance matrix is an N:N matrix that describes the similarity of every point in the data when compared to every other point. Generally, points close together on the X axis will be deemed more similar than others. This is a core component of the model and describes the overall trend of the resulting function. The covariance matrix is calculated by a covariance function, or kernel function. There are many kernel functions available that all make different assumptions as to what data points are deemed similar, and in turn define the shape of the resulting predictive functions. While the Gaussian process itself is non parametric, a kernel function has parameters that are often estimated through learning, and give further influence on the smoothness of the predictive function. The covariance matrix must be positive definite, i.e all entries must be positive otherwise the model collapses. One of the most common kernel functions is the Gaussian kernel (sometimes called the squared exponential) that is notably good at modeling a smooth function.

Traditionally the Gaussian process model has not dealt well with multiple outputs due to the complex nature of kernel functions. It is difficult to construct a positive definite matrix from one function that takes in many combinations of values, and in turn can accurately model the relationships between data-points. In 2005 Boyle et al[11] developed an extension of the Gaussian process regression model to handle mul-

tiple outputs. They used a Gaussian auto-covariance function and cross-covariance function to describe covariance within and between each data-point in each output:

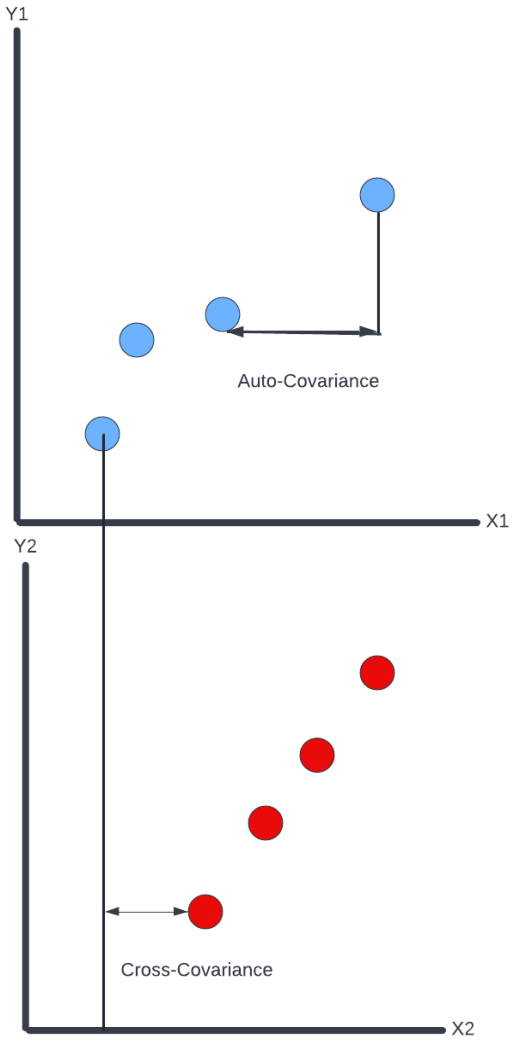


Figure 1. Diagram Illustrating the concept of auto and cross covariance. Two hypothetical datasets are shown in blue and red. Auto-covariance represents similarity between points within a dataset and cross-covariance represents similarity between the two data-sets.

The result of this is a covariance matrix containing both auto-covariance and cross covariance to accurately model more than one output. Boyle tested the predictive capability of their model when handling time series forecasting, and found it to be more accurate than a single output model when predicting sim-

ulated data. The data they simulated was made to be strongly dependant, where one time series was a lagged version of another. However boyles method was not tested on real world data which is likely to be more sporadic and noisy.

The aims of this proof of concept are to construct the model described by Boyle, and test it on simulated data and real continuous monitoring blood glucose data. Boyle's model will be compared with a single output Gaussian process, using the same Gaussian kernel function to asses the models viability when applied to CGM data.

## Results

### Simulated Data

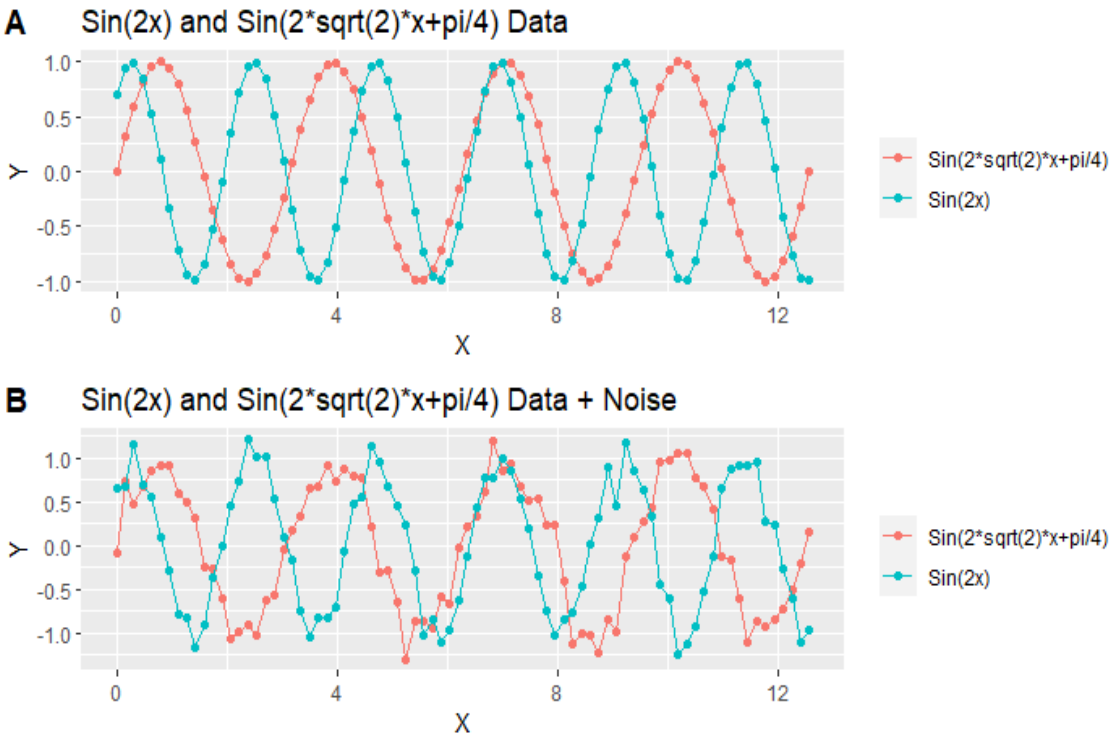


Figure 2. Simulated sine data. X shows each simulated time point and Y shows the output of the respective function. Each point and connective line is coloured according to the function. **(A)** Data with no Gaussian noise.**(B)** Data with added Gaussian noise $\mu = 0$ and $\sigma = 0.15$

fig.2 shows the simulated data with and without noise. The data follows a clear sine function trend, and it can be clearly seen that the function in red is a shifted version of the function in blue. The noisy data is more jagged, and therefor more realistic than that shown in 2(A).

The multiple output and single output GPs were ran on the noisy data shown in fig2(B).

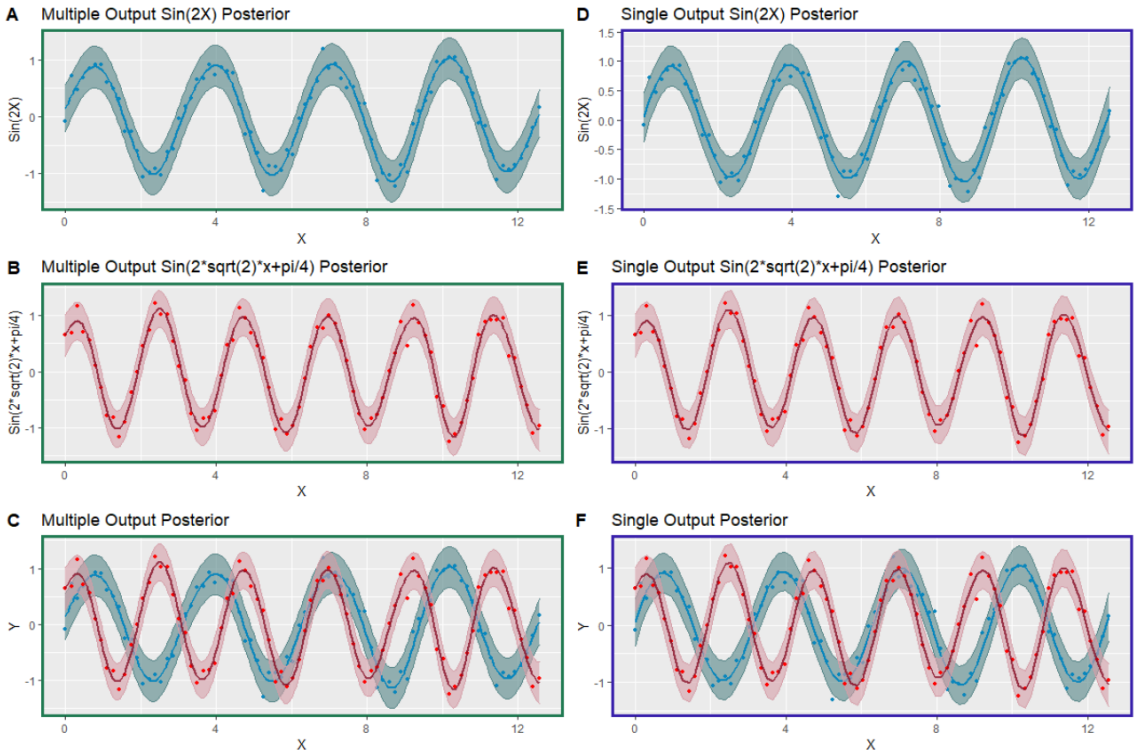


Figure 3. Posterior outputs of single and multiple output Gaussian processes on simulated time series. Solid lines show the mean function of the Gaussian process and ribbon shows the confidence interval at each point. X shows each simulated time point and Y shows the output of the respective function. **(A)** Multiple output Gaussian process performed on sin(2x) data. **(B)** Multiple output Gaussian process performed on $\sin(2 * \sqrt{2}x + \pi/4)$. **(C)** Overlay of Multiple output posteriors. **(D)** Single output Gaussian process performed on sin(2x) data. **(E)** Single output Gaussian process performed on $\sin(2 * \sqrt{2}x + \pi/4)$. **(F)** Overlay of Single output posteriors.

Fig.3 indicates that the multiple output GP fit a sine function to both instances of the data, showing

that the model is functioning as required. This can be seen when comparing fig.3(C) to fig.2, the GP shows a similar curve. Both of the models show higher confidence in the fitted function for the second data-set than they do the first, where very few data-points are intercepted directly by the mean function. The confidence intervals collapse where a data-point is directly intercepted by the mean, however it seems there are not many cases of this in either of the two sets of outputs.

In order to test the predictive capability of the models, the final 10 datapoints were removed and then predicted.

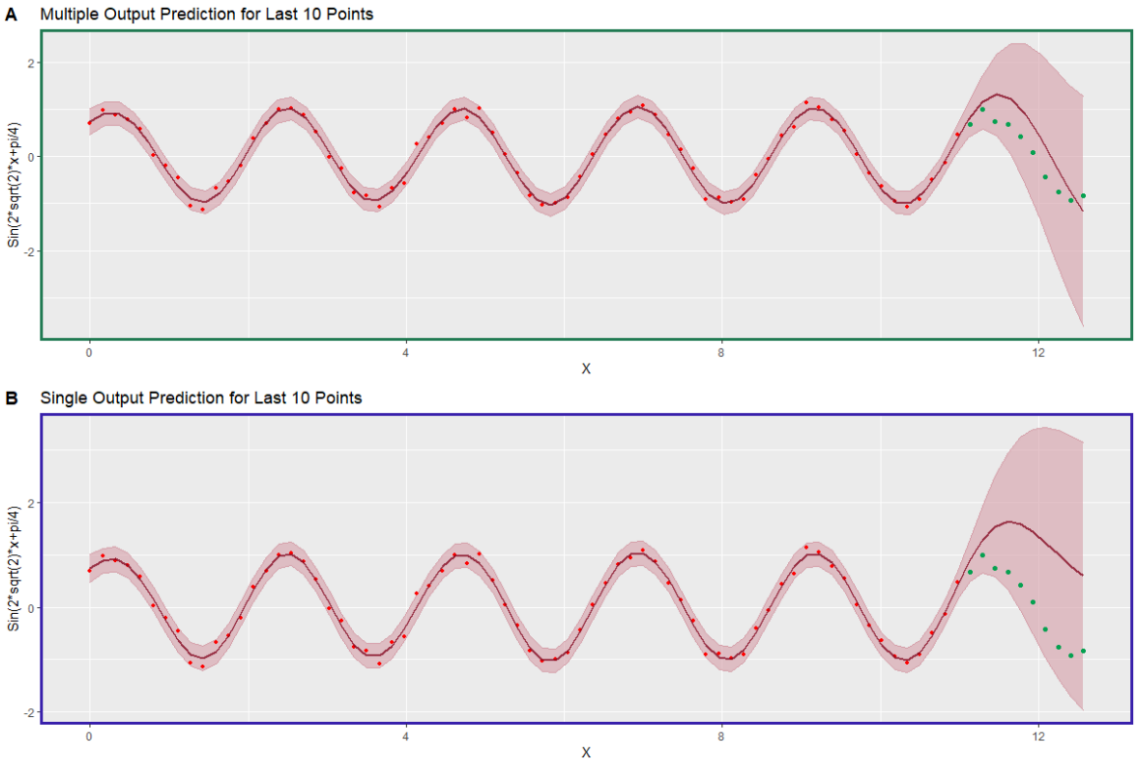


Figure 4. Prediction of final 10 datapoints in simulated data using multiple and single output GP. Solid red lines indicate the mean function, ribbon indicates confidence intervals. Green points show the ground truth. **(A)** Posterior output for multiple output model. **(B)** Posterior output for single output model.

It is clear that the multiple output model shows a more accurate forecast of the final 10 points when compared to the single output model. Although the mean function itself was near the points, the confi-

dence intervals were still almost as large as in the single output model. This indicates a large amount of uncertainty in predictions made in that region. In both models there is a downward trend in the mean function after no more data-points are present.

## Continuous Glucose Monitoring Data

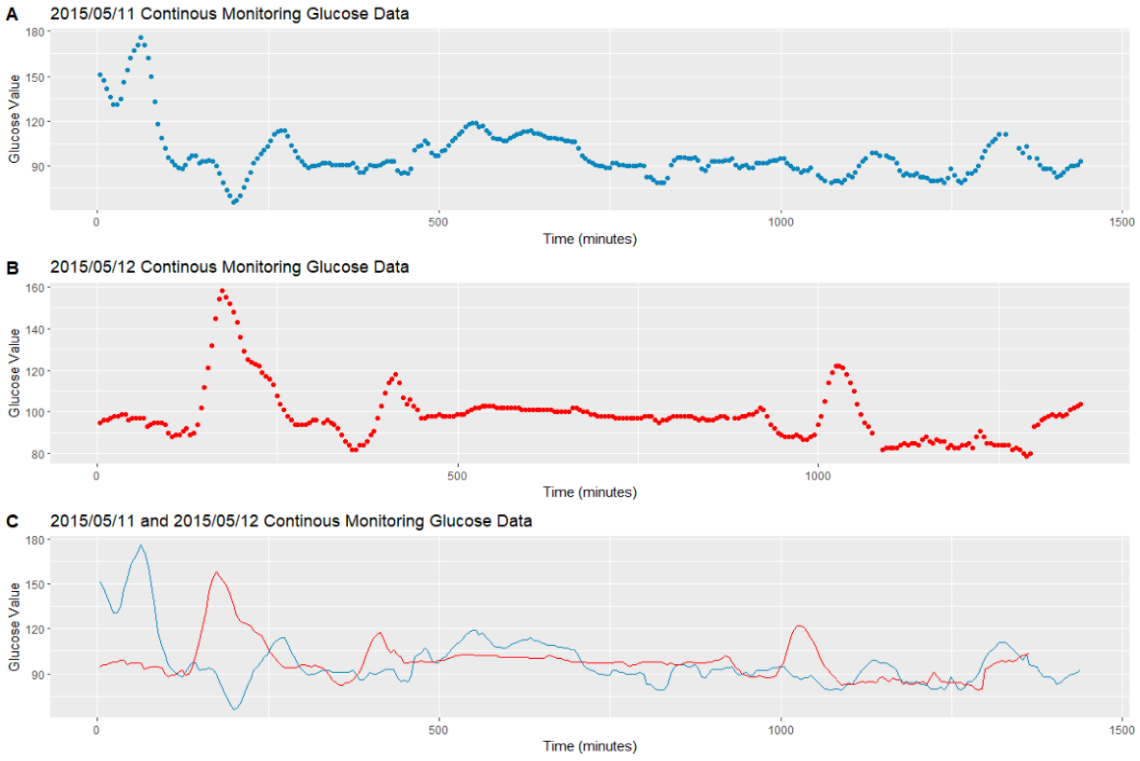


Figure 5. Blood glucose monitoring data from healthy patient id 1636-69-053 taken over two consecutive days. X shows the time in minutes and Y shows the blood glucose value. (A) Data from 2015/05/11. (B) Data from 2015/05/12. (C) Overlay line plot of both days

Fig.5 shows that data from the two days follow a similar trend, with a spike at the start and then a slow tapering off. The data from 2015/05/11 shows slightly more variation than the subsequent day. It should also be noted that the number of data-points is the same in both days, however sampling in the first day was further apart at some time points, and as a result 2015/05/11 ends later than 2015/05/12. Most data-points in both of the days are tightly clustered together.

As before, the single output and multiple output models were constructed on this data and the outputs inspected.

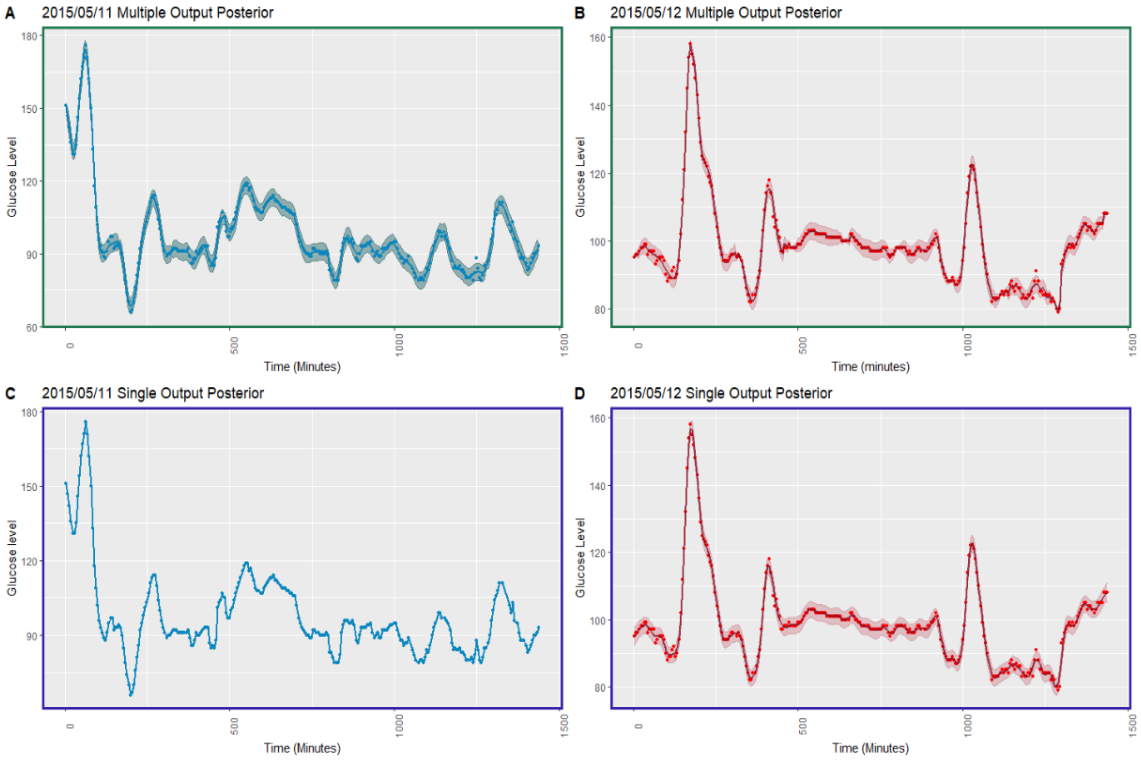


Figure 6. Outputs from multiple output and single output GPs. Solid line indicates the mean function and ribbons indicate confidence intervals at each point. X axis shows the time in minutes and Y shows blood glucose levels. **(A)** Multiple output posterior for date: 2015/05/11 **(B)** Multiple output posterior for date: 2015/05/12 **(C)** Single output posterior for date: 2015/05/11 **(D)** Single output posterior for date: 2015/05/12

Overall, the multiple output model shows more uncertainty on the first day; however, on the second day both models appear almost identical in their prediction of the mean and confidence intervals. The mean functions predicted almost directly match the line plot in Fig5(A) The final 10 points for day 2015/05/12 were predicted.

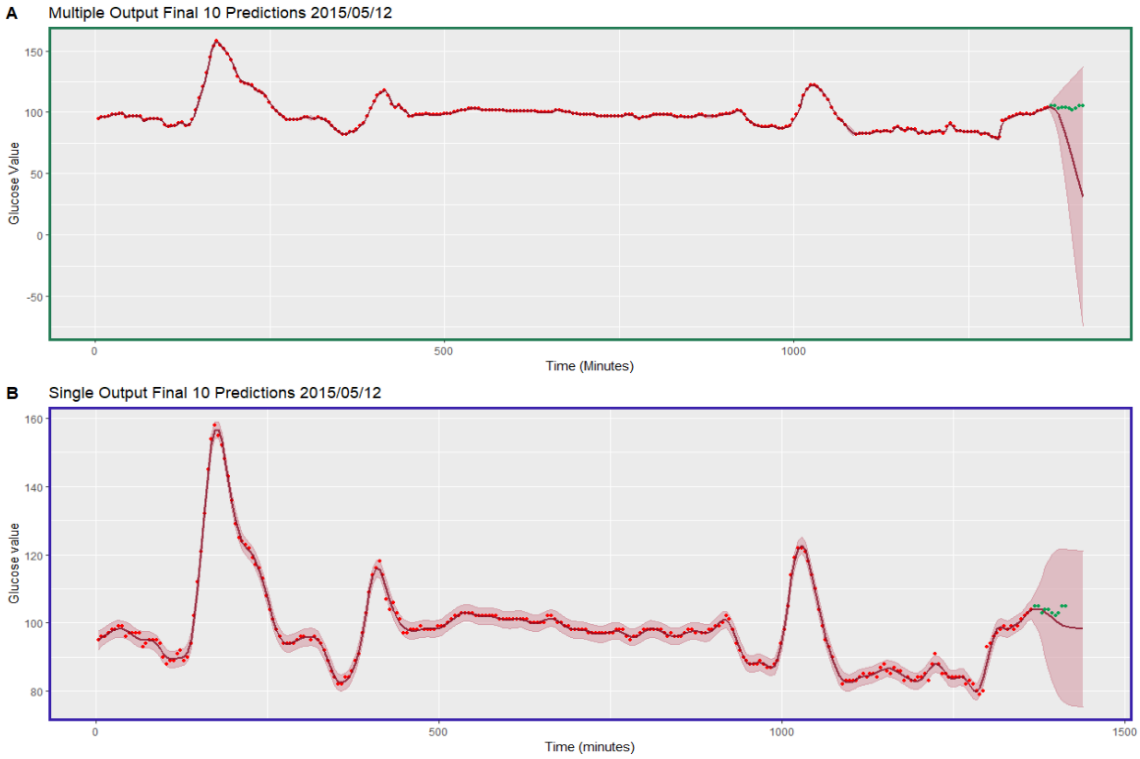


Figure 7. Outputs from multiple output and single output Gaussian process regression when predicting final 10 data-points. Solid line indicates the mean function and ribbons indicate confidence intervals. X axis shows time in minutes and Y axis shows blood glucose value. The ground truth is shown in green. **(A)** Multiple output posterior for date: 2015/05/12 **(B)** Single output posterior for date: 2015/05/12

Fig.7 shows a significant difference between the predictions. The multiple output model was inaccurate, with a large lower bound on its confidence interval. Its mean function trends downwards towards 0, wheras the ground truth implies a more stable function. On the other hand the single output model showed better accuracy, with the mean function covering some, but not all of the points. The predictive function is not ideal and the uncertainty was also very high for this output.

## Discussion

The overall goals of this paper were to reproduce Boyles model and to validate it on both simulated data and CGM data as a possible method to predict blood glucose. The initial testing of the constructed model on the simulated data was promising; the multiple output model was more effective when predicting the simulated data when compared to the single output model. This is likely due to the fact that it successfully captured the strong relationship between the two outputs, as opposed to looking at them as individual entities. This suggests that the implementation of the multiple output model works as it should. The model also managed to predict the underlying sine function that the data was built upon. This can be seen when comparing fig.2 to fig3, where the mean function predicted by the model was very similar to the line plots.

When running on all the blood glucose data, the model outputs in both cases were very similar, however the multiple output model showed a higher degree of uncertainty, with larger confidence intervals. This is likely due to the fact that it takes both outputs into account when fitting the mean function, leading to a larger range of possible predictions that have a lower overall probability. The multiple output model failed almost completely when predicting the last 10 data-points, with the single output showing a much more realistic mean function. It would appear that Boyle's method may be ideal for an artificial situation in which two outputs are strongly linked, however when dealing with noisy real world data the model struggles to accurately represent the relationship between outputs. Since the data used in this study is from a healthy individual, use of this method in a diabetes patient may be even less effective due to the likelihood of highly fluctuating data-points that may further perturb the predictive functions. One possible cause of this is the chosen kernel function for covariance and cross covariance. In particular, the Gaussian kernel used in this case generally fits a smoother function than other kernels[12]. If the function is smooth,

then this is modelling relationships between data-points that are further away, and not suitably modelling data that is close together. The simulated data that was used had a smooth underlying function with little noise (fig.2), which is likely the reason why the multiple output model fared better when making predictions with a Gaussian kernel. On the other hand fig.5 clearly shows that the blood glucose data is more jagged and noisy, with many clusters of points close together. Using the Gaussian kernel may therefor be the reason why the multiple output model performed poorly on this data.

Naturally, other kernel functions must be considered. One such possibility is a method of combining kernels. For example the Gaussian kernel could be combined with a less smooth kernel to accurately model more relationships between data-points both close together and far away. One such paper illustrates this, with a combination of the squared exponential kernel and matern 3/2 kernel to model a more jagged function over chemical data captured from an iron mine[13]. Kernel functions such as these may be better suited to the prediction of blood glucose levels. However it may be difficult to construct multiple output kernels that also function as a combination of two independent kernel functions and still give a positive definite output. More work will need to be done to test this.

It should also be observed that the parameters used in the kernel functions in the multiple output model were sampled each time from the same normal distribution: $\mu = 0$ and $\sigma^2 = 3$. Although this defines the prior assumption as to the nature of the data, the range from which parameters were selected during optimisation may have been too limiting. It may have been advantageous to define a wider distribution for the parameter selection, leading to parameters that were better optimised. The Gaussian kernel itself has parameters that scale the length and therefor smoothness of the relating function, so better optimised parameters could lead to better predictions. This is likely the reason why the single output model was more successful; the Gaupro package contains a more complex and effective way of optimising the parameters as opposed to the more basic Nielder Mead method used in the multiple output model. An example of

another possible method of optimisation is gradient descent which is widely used throughout machine learning.

The inefficiency of the multiple output model led to a significantly longer run time than that in the Gaupro package so more work needs to be done in order to improve optimisation and the overall efficiency of the multiple output model. In particular it may have been beneficial to construct the multiple output model by altering a package such as Gaupro or using a more versatile package such as RStan[14] for optimisation. A method such as cross validation would allow for the comparison of different prior distributions to assess how these affect the model, although this will be computationally expensive. This would improve the accuracy of the model.

Another factor at play is the number of consecutive time series used. In this study the multiple output model was used to fit two time series. More accurate predictions may be made if more consecutive days were used. For example a multiple output GP may be more accurate if it has more dependant data feeding into the model. In particular, the lagged time series Boyle simulated used three outputs in predictions, leading to greater accuracy than the independent model they compared it to. Using more than two days from the CGM data may also improve the prediction accuracy. It is likely that two consecutive days were not enough to thoroughly capture how the days relate, as the glucose data is far more complex than the two basic functions used to simulate the first data-set. Two days may not be enough to give an indication of events such as meal times or exercise which can significantly alter blood glucose concentration. Adding more ouputs is likely to increase run time and be more computationally expensive.

In summary the multiple output model requires more work in order to adapt it to the nature of blood glucose data. Further work would involve multple additions to the model. First, a better auto and cross covariance kernel must be defined and tested on the same data to assess if the kernel function is causing the poor prediction. Further testing should also involve the use of a larger set of consecutive CGM data,

ideally 3 or 4 days. Finally, more work needs to be performed on the optimisation algorithm and selection of parameters.

## Methods

### Multiple Output Gaussian Process Regression

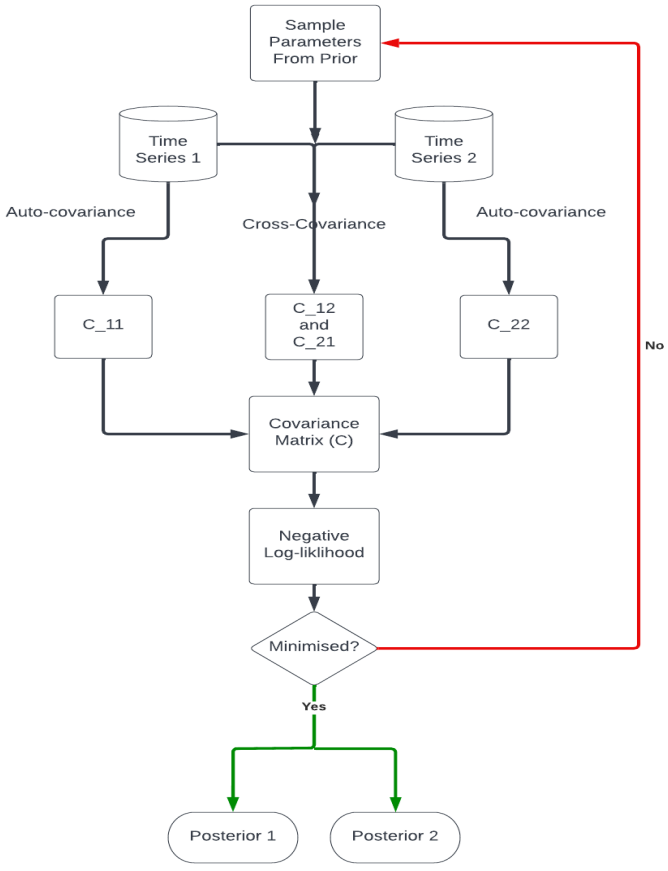


Figure 8. Schematic showing the multiple output Gaussian process for two data-sets. Auto and cross covariance is calculated which in turn allows for calculation of the negative log likelihood. This process is repeated with re-sampled parameters until the negative log likelihood is minimised. Posterior outputs from the model have the format of a mean and variance/confidence interval

The multiple output GP was constructed using Rlang (version 4.2.1) in Rstudio (version 1.1.546.). It was coded according to the components specified in Boyle's paper[11]. A Gaussian kernel was constructed to calculate the auto-covariance within each output ($C_{ii}, C_{jj}$) and cross-covariance between each output ($C_{ij}, C_{ji}$) for each data point $d$:

$$C_{ii}^{U}(d) = \frac{\pi^{\frac{p}{2}} v_i^2}{\sqrt{|Ai|}} exp(-\frac{1}{4} d^T A_i d)$$

$$C_{12}^{U}(d) = \frac{(2\pi)^{\frac{p}{2}} v_1 v_2}{\sqrt{|A_1 + A_2|}} exp(-\frac{1}{2}(d-\mu)^T \sum (d-\mu))$$

$$C_{21}^{U}(d) = \frac{(2\pi)^{\frac{p}{2}} v_1 v_2}{\sqrt{|A_1 + A_2|}} exp(-\frac{1}{2}(d+\mu)^T \sum (d+\mu))$$

$$C_{ii}^{V}(d) = \frac{\pi^{\frac{p}{2}} w_i^2}{\sqrt{|A_i|}} exp(-\frac{1}{4} d^T B_i d)$$

The covariance functions are parameterised by: $[v_1, v_2, w_1, w_2, A_1, A_2, B_1, B_2, \mu, \sigma_1, \sigma_2]$

Each parameter was assumed to derive from a Gaussian distribution with $\mu = 0$ and $\sigma^2 = 3$. This coincides with the baysian method of defining a prior assumption about the data. The parameters were sampled from this distribution using the rnorm function in R.

The combined outputs of each kernel function generate a covariance matrix $C$ which captures the relationships between every data-point:

$$C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

This covariance matrix contains the 4 covariance matrices generated by the functions above. Once a covariance matrix is produced the negative log likelihood can be calculated as follows

$$l = -(-\frac{1}{2} log|C| - \frac{1}{2} y^t C^{-1} y - \frac{N_1 + N_2}{2} log 2\pi)$$

In this case a negative log likelihood was used. Learning of the optimal parameters therefor involves minimisation of the negative log liklihood. This was performed using the Neilder Mead algorithm available in R using the optim function.

Finally, posterior predictions were made as follows, where the predictive distribution has mean $\hat{y}$ and variance $\sigma_y^2$

$$\hat{y} = K^T C^{-1} y$$

$$\sigma_y^2 = k - K^T C^{-1} K$$

where:

$$k = C_{ii}^Y(0) = v_i^2 + w_i^2 + \sigma_i^2$$

$$K = [C_{i1}^Y(s^! - s_{1,1})..C_{i1}^Y(s^! - s_1 N_1) \quad C_{i2}^Y(s^! - s_{2,1})..C_{i1}^Y(s^! - s_2 N_2)]^T$$

The Posterior outputs were visualised using GGplot2[15]

### Single Output Gaussian Process

In order to assess the efficacy of Boyles model, a single output Gaussian process was also ran on each time series for comparison. The single output model works as shown in fig.8, except without the use of a cross covariance function. Two independant models were therefor ran for each set of two time series. The single Output GP was implemented in R using the Gaupro package (version 0.2.4)[16]. In all cases the default settings were used alongside the default Gaussian kernel which is the same as $C_{ii}$ above.

### Simulated Data

In order to confirm that the model is working as it should and to confirm the findings of Boyles paper two simulated time series were generated. These consist of 80 data-points with range 0 to $\pi \times 4$. Y values

were generated using two versions of the same function:

$$Y_1 : \qquad Y_1 = sin(2X)$$

$$Y_2 : \qquad Y_2 = sin(2 * \sqrt{2}x + \pi/4)$$

$Y_2$ is simply a shifted version of $Y_1$, and as a result these two time series show a strong dependant component. Gaussian noise with $\mu = 0$ and $\sigma = 0.15$ was added to each data point iteratively in order to make the simulated data more representative of data seen in the real world. Both the multiple output and single output models were ran on this data. Predictions were made for the final 10 data points of $Y_2$ and accuracy of the mean functions was compared between the two models.

### Continuous Glucose Monitoring Data

A publicly available data-set published by Hall et al[17] was used for blood glucose prediction. The data-set consists of blood glucose recordings from 57 patients across 3 years. Recordings were taken at roughly 5 minute intervals each day, although this varies from patient to patient. The individuals are a mixture of healthy and diabetic patients. For this proof of concept, a healthy individual was chosen with less sporadic blood glucose levels and a consistent rate of blood glucose readings; the patient was ID 1636-69-053. The data was loaded and preprocessed in R (version 4.2.1). The internal time column was used as X, and times were converted from hour format to minutes in order to allow the models to process the values as inputs. All data was graphed using ggplot and two consecutive days of blood glucose readings were chosen: 2015/05/11 and 2015/05/12 which were ran through the multiple output model as dependant outputs and through two independent Gaussian process regression models. The final 10 data points of

day 2 were predicted and compared to assess the accuracy of the multiple output model. The data and study used can be accessed here (denoted as S1 table in the study).

### System Specifications

All analysis was ran on a windows 10 desktop with the following technical specifications: Intel I7 processor, NVIDIA GTX1070 graphics card, 16GB RAM, 500GB ssd hard drive.

## References


1. Arditi, C., Zanchi, A. & Peytremann-Bridevaux, I. Health status and quality of life in patients with diabetes in switzerland. *Prim. care diabetes* **13**, 233–241 (2019).

2. You, W.-P. & Henneberg, M. Type 1 diabetes prevalence increasing globally and regionally: the role of natural selection and life expectancy at birth. *BMJ open diabetes research care* **4**, e000161 (2016).

3. Cobelli, C., Renard, E. & Kovatchev, B. Artificial pancreas: past, present, future. *Diabetes* **60**, 2672–2682 (2011).

4. Marling, C. & Bunescu, R. The ohiot1dm dataset for blood glucose level prediction: Update 2020. In *CEUR workshop proceedings*, vol. 2675, 71 (NIH Public Access, 2020).

5. Xie, J. & Wang, Q. Benchmark machine learning approaches with classical time series approaches on the blood glucose level prediction challenge. In *KHD@ IJCAI* (2018).

6. Zhou, Y., Chang, F.-J., Chang, L.-C., Kao, I.-F. & Wang, Y.-S. Explore a deep learning multi-output neural network for regional multi-step-ahead air quality forecasts. *J. cleaner production* **209**, 134–145 (2019).

7. Li, K., Liu, C., Zhu, T., Herrero, P. & Georgiou, P. Glunet: A deep learning framework for accurate glucose forecasting. *IEEE J. Biomed. Heal. Informatics* **24**, 414–423, DOI: 10.1109/JBHI.2019.2931842 (2020).

8. Dumitru, C. & Maria, V. Advantages and disadvantages of using neural networks for predictions. *Ovidius Univ. Annals, Ser. Econ. Sci.* **13** (2013).

9. Rasmussen, C. E. *Gaussian Processes in Machine Learning*, 63–71 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2004).

10. Roberts, S. *et al.* Gaussian processes for time-series modelling. *Philos. Transactions Royal Soc. A: Math. Phys. Eng. Sci.* **371**, 20110550 (2013).

11. Boyle, P. & Frean, M. Multiple output gaussian process regression. (2005).

12. Wilson, A. & Adams, R. Gaussian process kernels for pattern discovery and extrapolation. In Dasgupta, S. & McAllester, D. (eds.) *Proceedings of the 30th International Conference on Machine Learning*, vol. 28 of *Proceedings of Machine Learning Research*, 1067–1075 (PMLR, Atlanta, Georgia, USA, 2013).

13. Melkumyan, A. & Ramos, F. Multi-kernel gaussian processes. In *Twenty-second international joint conference on artificial intelligence* (2011).

14. Stan Development Team. RStan: the R interface to Stan (2020). R package version 2.21.2.

15. Wickham, H. *ggplot2: Elegant Graphics for Data Analysis* (Springer-Verlag New York, 2016).

16. Erickson, C. *GauPro: Gaussian Process Fitting* (2021). R package version 0.2.4.

17. Hall, H. *et al.* Glucotypes reveal new patterns of glucose dysregulation. *PLOS Biol.* **16**, 1–23, DOI: 10.1371/journal.pbio.2005143 (2018).