

RADBOUD UNIVERSITY NIJMEGEN



FACULTY OF SCIENCE

Predicting Depression with Bayesian Nonparametric Models

THESIS MSc COGNITIVE COMPUTING

Author:
Zuzanna FENDOR

Supervisor:
dr. Max HINNE

Second reader:
dr. Luca AMBROGIONI

August 2022

Contents

1	Introduction	2
1.1	Aim of the project	3
2	Background	3
2.1	The course of depression	3
2.2	Data	4
2.3	Venlafaxine and its effects	5
3	Methods	5
3.1	Notation	5
3.2	More on the Network Approach and Graphical Gaussian Models	6
3.3	MGARCH	7
3.4	Bayesian nonparametric methods	8
3.4.1	Gaussian processes	8
3.4.2	Kernel functions and mean functions	8
3.4.3	Multi-output Gaussian Processes with Coregionalisation	10
3.4.4	Coregionalization implementation	10
3.4.5	Generalized Wishart Processes	11
3.4.6	Training the model	12
3.4.7	BANNER and scalability	15
3.5	Data processing	15
3.6	Data simulations	16
3.7	Experiments	17
3.8	Evaluation	17
4	Results	19
4.1	Simulation data study	19
4.2	ESM data study	23
4.3	Network results	27
4.4	Performance observations	29
5	Conclusion	29
6	Discussion	30
7	Acknowledgements	31

Abstract

The network model approach is a fairly recent paradigm for modeling psychopathological conditions like Major Depressive Disorder. This approach models the disorders as an interconnected network of symptoms. The connections in such a network are usually assumed to be static, however, there are reasons to disregard this assumption. In this project, we try to apply a Bayesian nonparametric method called the Generalized Wishart Process (GWP) to model the symptom connections in the form of a dynamic covariance matrix. We want to investigate whether this method is capable of learning such dynamic network structures and whether adding the dynamics helps the model to make better predictions. The GWP was able to learn the dynamic network structures, although it was worse at it than the parametric baseline with a dynamic covariance. The GWP was also worse at predicting future symptom values than another Bayesian nonparametric model that used static covariance. Nevertheless, it showed promising results in terms of interpretability as compared to the baseline, a property valuable in model prediction.

1 Introduction

Major depressive disorder (MDD) can be broadly defined as “a distinct change of mood, characterized by sadness or irritability and accompanied by at least several psychophysiological changes, such as disturbances in sleep, appetite, or sexual desire; constipation; loss of the ability to experience pleasure in work or with friends; crying; suicidal thoughts; and slowing of speech and action. These changes must last a minimum of 2 weeks and interfere considerably with work and family relations” [1]. It is estimated that 2% to 21% of the world population will have MDD during their lifetime, as shown in a recent systematic review of the disorder [2]. The social burden is even larger if we consider the effect MDD can have on other people, like the family, friends, and coworkers of a person with depression. Despite being so prevalent, the exact mechanisms of the disorder are still not fully known [1].

The disorder can have different root causes and its course can vary from individual to individual, with different responses to treatment. Moreover, some people seem to be inherently more vulnerable to depression and might become depressed as an effect of situations that would not leave a lasting effect on other people[1][3].

In the past, mental disorders were often modeled using latent variable models, where the mental disorder was seen as the single cause of the symptoms. The symptoms themselves are observable and can change over time, however, according to this view they are independent given the disorder.

In a recently proposed perspective, MDD is instead represented by the interactions between measurable symptoms. This *network model* perspective adds the relationships between the symptoms on top of the already present individual symptom dynamics. The network representation offers an opportunity to describe the disorder in a more informative manner. For instance, the network model can be used to explain why some individuals are more prone to developing MDD than others [3]. It also has some important consequences for the diagnosis and treatment of MDD. For example, if the connections of the symptoms are known, MDD could be treated by targeting the most influential symptoms in the network[3]. The challenge lies in estimating the symptom networks.

A common approach that can be used for modeling such networks based on psychological data is the Gaussian Graphical Model (GGM) [4]. Those models estimate the covariance matrix by approximating the multivariate Gaussian distribution of the random variables given the data. The covariance between variables describes their joint variability. In other words, a positive covariance means that when one variable varies in one direction, the other variable varies in the same direction. It is a measure of the relationship between the variables, although it is not a normalized one. The covariance matrix estimated with GMMs is static and the model itself is not designed to model and predict an input-output variable relationship in a time series. Due to this fact, the regular GGM cannot be used to model dynamic networks without an additional tool like window-wise estimation, where the data is divided into windows.

As noted by Cramer et al. [3], it might be advantageous to have a symptom network that can change over time. Testing this suspicion is technically challenging because estimating the dynamics of a network in series, here called a network regression problem, does not have many existing solutions yet [5]. Simply grouping the data based on the dosages in a sliding-window manner does not always work well, because it is influenced largely by the window length, offset, and filtering [6]. Existing frameworks that try to explicitly capture the dynamics of the networks struggle with irregularly sampled data, scalability, and interpretability of the parameters [7]. To address these problems we propose a Bayesian nonparametric solution based on the Generalized Wishart process [8] that builds upon Gaussian process regression [9]. The Generalized Wishart process estimates a dynamic prior on the covariance matrix, allowing for network dynamics. At the same time, Gaussian processes can use the relationships between the outputs to inform better predictions. In theory, this combination could be beneficial for predicting MDD symptoms, but to the best of our knowledge, it has not yet been applied to this or a comparable task at the time of writing.

1.1 Aim of the project

The long-term goal of this project is to provide more understanding of how the MDD-related symptom networks change depending on the anti-depressant dosage. This requires an accurate way of capturing the relationships and dynamics of the symptoms. For this reason, the focus of this project lies in further developing and testing Bayesian nonparametric approaches to estimating dynamic networks. Moving in this direction will hopefully help us better describe the disorder in line with the new network model perspective and produce better predictions for the symptoms.

Ideally, this project will provide new insights into the field of modeling mental disorders. A better model of how various symptoms interact and change over time has the potential for enabling better, more personalized treatment plans for the patients in the future.

Finally, the AI field might profit from the insights and implementational choices that will be accumulated during this project. Capturing network dynamics is a difficult problem in machine learning. Predicting MDD with dynamic network structures using a Bayesian nonparametric approach can serve as a proof of concept for other machine learning models.

The two research questions are:

- *Can observed symptoms of MDD be better predicted when explicitly capturing the dynamics of their interactions?*
- *Can we capture the properties of MDD in terms of covariance functions and mean functions in a nonparametric Bayesian model?*

2 Background

2.1 The course of depression

As stated before, the course of depression is very individual-specific. Nevertheless, knowing some general trends may help initialize the model correctly.

According to the DSM5 [10], major depressive disorder is characterized by discrete episodes of at least 2 weeks' duration. During this period, the person experiences changes in affect, cognition, and neurovegetative functions. When the depressive disorder persists longer than 2 years in a row for adults or longer than 1 year for children, it gets categorized as dysthymia. While dysthymia is a kind of depression that is not entirely the same as MDD, it includes the same list of symptoms. For this reason, we will not make any assumptions about the maximal length of symptoms for MDD symptoms.

In general, a depressive episode can be subdivided into six main phases: episode, remission, response, recovery, relapse, and recurrence (Table 1). It seems that the majority of people recover within one year[11], with an average of four to six months[12]. However, there are also people for whom an episode can last many years.

The length of the episode is also a predictor of the recovery rate. Longer episodes are generally harder to treat. According to the NIMH Collaborative Depression Study study as summarised by Richards [11], the probability estimate of the rates of recovery after one year was 67%. It was 81% recovery by 2 years, by 5 years 88% recovery and by 10 years 93%. Furthermore, the second, third, and fourth episodes did not differ significantly from the first episode in terms of recovery rates, meaning that a person suffering from their second, third or fourth depressive episode is not significantly less likely to recover. The recovery in the fifth episode was notable yet still not significantly slower. These findings are based on a clinical setting, however, Richards also mentioned another study based on the ECA general population survey that seems to point in the same direction. Moreover, the ECA study showed that about 50% of participants had no future depressive episodes after the first one. About 14 to 35% suffered from a recurrent depressive episode during one year and 6 to 15% suffered from ongoing depression. For more details about the source of those statistics we refer the reader to Richards [11] and the studies cited in this review.

Key Terms	Definition
Episode	Defined as having a certain number of symptoms for a certain period of time, fully symptomatic. (e.g. DSM-IV criteria, see Classification)
Remission	Partial remission where the individual is no longer fully symptomatic, but displays more than minimal symptoms. Full remission is a brief period (2–8 weeks), where the individual is asymptomatic, no more than minimal symptoms
Response	A partial or full remission due to a treatment intervention
Recovery	Defined as a full remission, symptom-free for a certain length of time (>8 weeks). It designates a recovery from an episode
Relapse	An early return of symptoms following a positive response, meeting full syndrome criteria that occurs during the period of remission
Recurrence	Refers to a new episode, which can only occur during a recovery

Table 1: The terms describing the key phases in depression and their meaning, as defined by Richards [11]

The remission of symptoms is not always full. Some patients experience only a reduction in some of the symptoms. It appears that partial remission is an important indicator of the chance of relapse [11] [13]. The study by Paykel found that 76% of participants with partial remission experienced a relapse as opposed to 25% rate of relapse in the group with full remission.

Of course, depressive symptoms are not all or nothing. People who do not suffer from major depressive disorder can experience depressive symptoms. Likewise, people who are depressed may not experience all of the possible symptoms at once or have some better days. However, we assume that in both cases, those can be considered outliers in general, with longer trends of at least two weeks.

We are aware that this section contains many simplifications and it should not be taken as generalizable ground truth for MDD. Rather, it is supposed to be a starting point for initializing models in an informed manner.

2.2 Data

The dataset [14] used in this project, henceforth called the ESM dataset, comes from a single participant ($n = 1$) with MDD who monitored their experience over the course of 239 consecutive days. While a

sample of one is in no way generalizable, there are to our knowledge few available studies providing such extensive longitudinal data. Moreover, the diverse expressions of depression make it unclear how the results should be aggregated, especially since the participants may differ in both the intensity and combination of symptoms, as well as the underlying symptom networks. While more general models are useful in their own right, the models based on individual experiences capture the trends that could be further used for personalized treatment plans and are by no means useless, For these reasons we kept to just this data set and focused the research on the more personal form of modeling.

The experiment the ESM dataset was based on included a double-blind phase in which the participant’s dosage of antidepressant venlafaxine was reduced according to a randomly chosen reduction plan. The ESM dataset consists of almost 50 items that were filled in several times a day, daily, or weekly. The topics asked about included: the current mood, the present company and the level of pleasantness of that company, self-esteem, physical condition, the activity the participant was doing before the assessment, and the last important event since the last assessment. Once a week, the participant’s depression symptoms were assessed using a shortened version of the SCL-90-R questionnaire. Since a network with 50 items is unreadable and difficult to analyze, we select only a few variables of interest. Taking inspiration from the study by Cramer et al. [3] and the work that this dataset originates from [14], the network will include: negative affect, positive affect, mental unrest, worrying, and feeling suspicious. Those variables will be constructed as summaries of one or several items from the original dataset. Since the dataset contains multiple variables in each category, the variables will be summarised using a mean of the scores. Some variables do not have true ordinal values, like the current social setting and the current event. We will omit all categorical variables. This reduces the complexity of the models and enables us to use the baseline model that requires numeric values.

The only explicit input variable will be time. The dosage of the antidepressant will not be known by the models.

2.3 Venlafaxine and its effects

Venlafaxine [15] is the antidepressant taken by the participant of the ESM study. It is a serotonin-norepinephrine reuptake inhibitor that is supposed to improve the brain chemistry of a depressed person to reduce fatigue and depressed mood and improve their interest in daily activities. The withdrawal symptom list for stopping taking venlafaxine is extensive and also partially overlaps with depression symptoms. Some of the possible symptoms of withdrawal are included in the daily questionnaire, namely: mood changes, agitation, anxiety, tiredness, dizziness, dry mouth, nausea, headache, appetite, and trouble falling asleep. According to the data description, the participant had to be put back on venlafaxine after the duration of the experiment where the dosage was reduced to zero, because of the recurrence of the symptoms.

However, was the intensity of the symptoms the only change due to the reduction of the venlafaxine dosage, or did the underlying cause-effect structure of the symptoms change as well? To our knowledge, the effects of venlafaxine on the strengths and directions of the connections are unknown. We can gain a bit of insight into a potential relationship by plotting the dynamic network structure for the different experimental phases, however, with a dataset of only one participant, it is not possible to rule out external factors and derive a relationship between the medication and the network model. Hence we use the different venlafaxine dose stages only as a convenient division of time when comparing our network structures to the models created by Wichers et al [16] based on the same data.

3 Methods

3.1 Notation

We denote our dataset as the combination of inputs X and outputs Y , where generally $X \in \mathbb{R}^N$ and $Y \in \mathbb{R}^{N \times D}$. N denotes the number of data points and D the number of output variables. In the case of the ESM dataset, X translates to the time points and Y to the symptom values. Since X refers to

the time, we will use the subscript t for indexing variables that change depending on the input X . This choice was made such that it is at all points clear to the reader that the variables are dynamic with respect to time. In practice, each model requires slight reshaping of the data. For instance, the GWP model requires X to be shaped as $D \times N$. However, the meaning of X , Y , N , and D remains the same. Moreover, some of the letters, for example B in the definition of VAR (Eq. (1)) and B in the definition of the coregionalization matrix (Eq. (15)) will have a different meaning. This is due to the shortages in the alphabet. We have tried to keep equivalent parameters across different formulas and methods consistent.

3.2 More on the Network Approach and Graphical Gaussian Models

The network approach for modeling in the field of psychopathology is still a broad term. It may include models constructed from the causal relations in the diagnostic criteria themselves. The diagnostic criteria often already contain clues about what symptom may cause the other. The second way is to ask clinicians and patients about their experience of the causality of the symptoms. Finally, and this is what this project is about, the network models can be extracted empirically from data using statistical means. The network model here will be correlational in nature, only capturing the strength of the connections, not their direction.

There is also a difference in the scale of what the model is trying to convey [17]. One can try to model the symptoms of one individual in a vacuum, without any input from the outside information. Models can be also drafted from many individuals to discover general trends in a population. Finally, there are also models like the extended psychopathology networks that are essentially networks representing different individuals interacting with each other, capturing how the symptoms of one person can affect the other person's symptoms. In this project, we will limit ourselves to network models of one person in a time-series study. This comes with a caveat. Namely, as mentioned in the introduction, currently the main method of creating those models from data is through Graphical Gaussian Models. Graphical Gaussian Models are usually constructed from cross-sectional data, where each subject is measured only once, and the individual data points are assumed to be independent [4]. This is in stark contrast with the time-series data, where each data point is strongly temporally correlated. It is still possible to obtain a GGM from time series data, using Vector Autoregression (VAR) [4] which can be seen as a generalization of GMM.

$$\mathbf{y}_t = \mathbf{B}\mathbf{y}_{t-1} + \boldsymbol{\epsilon}_t \quad (1)$$

$$\boldsymbol{\epsilon}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}) \quad (2)$$

VAR provides us with both a temporal, directed network indicating the causality through B and a contemporaneous network, which is undirected and described by the variance-covariance matrix Σ . This project focuses on undirected, correlational modeling, therefore Σ would, in our case, be the component of interest.

GMMs and graphical VARs are based on partial correlation network structures.

$$pcor_{i,j} = \frac{\Sigma_{i,j}^{-1}}{\sqrt{\Sigma_{i,i}^{-1}} \sqrt{\Sigma_{j,j}^{-1}}} \quad (3)$$

Eq. (3) is essential, as it shows the relationship between the covariance, estimated by the models in this project, and the final network model which is the partial correlation network. Partial correlations are normalized to the scale of $[-1,1]$ and they factor out indirect relations, such that when A is related to C through B, there will only be direct links (A-B) and (B-C), and there will be no link (A-C). This helps us discover the graph of the direct relations between the symptoms [17].

In this project, we will try 3 alternative methods of approximating such graphs. The methods themselves estimate the covariance which then can easily be turned into a partial correlation.

3.3 MGARCH

MGARCH stands for Multivariate Generalized AutoRegressive Conditional Heteroskedasticity. As the name suggests, models the heteroskedasticity of the data; the change in the (co-)variance over time. MGARCH is commonly used as a model of volatility in financial data. It often poses as a baseline for new models that try to capture similar processes involving dynamic volatility of the data [9, 18, 19]. For this reason, we will be using MGARCH in this project to establish a performance standard. MGARCH knows different flavors. The basic framework of MGARCH is formulated as

$$y_t = \Sigma_t \eta_t \quad (4)$$

where Σ_t is a $D \times D$ matrix representing the conditional covariance at time t , conditioned on all information up until $t - 1$. η_t is the white noise process at t with the expectation $\mathbb{E}[\eta_t \eta_t^\top] = I$. y is a zero mean stochastic vector process that in our case describes the symptom outputs. The way of defining Σ conditioned on past information is the core of this method and at the same time, it is where the different variants of MGARCH deviate. Silvennoinen [7] distinguishes between four classes of MGARCH models: models that directly model Σ , factor models that assume that y is generated by a number of heteroskedastic factors, models that describe the conditional variance and correlation, and finally (semi-)nonparametric MGARCH models.

In this project, we chose the Dynamic Conditional Correlation (DCC) MGARCH model as our parametric baseline. This version is a member of the third category of MGARCH models. DCC-MGARCH was chosen because of its widespread use and its similarity to the multi-output coregionalization Gaussian process, which will be further explained in Section 3.4.3

The conditional covariance is estimated by [7]:

$$\Sigma_t = \mathbf{B}_t \mathbf{P}_t \mathbf{B}_t \quad (5)$$

$$[\Sigma_t]_{ij} = \sigma_{it}^{1/2} \sigma_{jt}^{1/2} \rho_{ijt}, \quad \text{for } 1 \leq i, j \leq D, i \neq j \quad (6)$$

where \mathbf{B}_t is the diagonal matrix $\mathbf{B}_t = \text{diag}(\sigma_{1t}^{1/2}, \dots, \sigma_{Dt}^{1/2})$ and \mathbf{P}_t is the correlation matrix with ρ_{ijt} being the correlation between elements dimensions i and j at time t , and with $\rho_{iit} = 1$. The individual conditional variances σ are often defined as univariate GARCH(p,q) models with [7]:

$$\sigma_t = \omega + \sum_{j=1}^q \mathbf{A}_j \mathbf{r}_{t-j} \odot \mathbf{r}_{t-j} + \sum_{j=1}^p \mathbf{B}_j \sigma_{t-j} \quad (7)$$

p, q determine the order of the GARCH. More exactly, they determine how many steps back of the stochastic process r and the variance σ are used to compute the next step at t . \mathbf{A}_j and \mathbf{B}_j are $D \times D$ diagonal matrices.

Having the correlation matrix be dynamic increases the flexibility of the model at the price of more costly computations, since now the correlation matrix needs to be inverted T times. Nevertheless, our dataset is relatively small making it possible to use the less restrictive assumptions on dynamics. The correlation matrix P itself is defined as [7]:

$$\mathbf{P}_t = (\mathbf{I} \odot \mathbf{Q}_t)^{-1/2} \mathbf{Q}_t (\mathbf{I} \odot \mathbf{Q}_t)^{-1/2} \quad (8)$$

$$\mathbf{Q}_t = (1 - a - b) \mathbf{S} + a \epsilon_{t-1} \epsilon_{t-1}^\top + b \mathbf{Q}_{t-1} \quad (9)$$

\mathbf{Q} is here a dynamic matrix process. a and b are scalar parameters for which $a + b < 1$, \mathbf{S} is a unconditional covariance parameter of the standardised errors ϵ_t . \mathbf{Q} is ensured to be positive definite at all times and it is made into a valid correlation matrix through rescaling in Eq. (9). \mathbf{P} being positive definite was one of the requirements of positive definiteness of Σ when defined by Eq. (5). The other requirement is for the variances σ_t to be well-defined, which is dependent on the univariate GARCH processes ruling them. The number of parameters for DCC-MGARCH is $(p + q + 1)D + (\bar{p} + \bar{q})$ [20].

While the basic framework assumes y to have a zero-mean, this does not necessarily have to be the case. To obtain the output predictions not centered around zero using MGARCH, all we need to do is use $Y_t | I_{t-1}(Y) \sim Z(\mu_t, \Sigma_t)$, where I_{t-1} is the information set of Y at time t and Z is any multivariate distribution. Since we assume a Gaussian distribution in all of the models we are using, Z will assume the form of a multivariate normal distribution. μ can be defined by different functions. The popular pairings with MGARCH are among others ARMA and VAR.

Our DCC-MGARCH implementation was built using the R package called `rmgarch`[21]. We used Vector Autoregression with lag 2 as the mean function. We use DCC order(1,1). The out sample predictions are made using the `RollForecasting` class. The test data is predicted in a rolling fashion and the model is refitted every 5 data points. For the historic predictions of the covariance, we used the `dccfit` function to fit the model and then extracted the covariance using the `rcov` function.

3.4 Bayesian nonparametric methods

3.4.1 Gaussian processes

This section is a short recap of Gaussian processes. For a more detailed explanation, we refer the reader to the book by Rasmussen and Williams [22] or the gentle introduction by Roberts et al. [23]. A Gaussian process is a generalization of the Gaussian probability distribution, where it provides us with a prior distribution over (properties of) functions instead of just a distribution of random variables. [22]. By combining that prior with the likelihood, we can obtain a posterior distribution over the function space. Formally, we define a Gaussian process as a collection of random variables, any finite number of which have a joint Gaussian distribution [22]. A Gaussian process is written as $u(x) \sim GP(m(x), k(x, x'))$, where $m(x)$ is the *mean function* and $k(x, x')$ is the *kernel function*. The mean function is the expectation of the functions, $\mathbb{E}(x)$. It determines the average trend of functions in absence of (nearby) data points. A zero mean function is often used to simplify the equations and to let the functions fully rely on the kernel. With the zero mean function, the average of all functions approaches the horizontal zero line. The mean function can be used when one has some external knowledge about the data, for instance, if there is already an empirical formula describing the behavior.

The kernel function describes the relations between the function values at different input points. Formally, we can say that $k(x, x') = cov(u(x), u(x'))$. The kernel determines the properties of the functions like their smoothness and periodicity. There are many different types of kernels, some popular kernels being the squared-exponential kernel, the exponential kernel, the periodic kernel, and more. Moreover, multiplication or a sum of two valid kernels is also a valid kernel, which means that the covariance matrices resulting from that kernel are always positive (semi-) definite and therefore also a valid kernel. This broadens the range of possible kernel functions even further.

3.4.2 Kernel functions and mean functions

Predicting depression is complex. Not all characteristics of the course of the disorder can be captured well by simply putting the data into a default Bayesian nonparametric model. Some important things to consider are long-range temporal correlations, periodicity, sudden changes in its course, and drifts of the symptoms. These are especially important when trying to make long-term predictions because, in the absence of data, those factors for a large part determine the behavior of the model. Bayesian nonparametric methods that use the Gaussian process can address those problems through mean functions, covariance functions, and their hyperparameters. Those functions can be crafted to capture certain characteristics, like the periodicity of the covariance. As noted previously, it is possible to combine multiple existing functions with certain characteristics at the cost of an increasing number of hyperparameters. Currently, the best fitting mean and covariance functions are not known for this problem. Finding the right functions is a challenge that needs to be tackled to make the predictive depression models perform to their full potential.

Three popular simple options for kernels are the squared exponential kernel, the Matérn kernel family

and the exponential kernel [23] [22]. They are defined in their respective equations Eq. (10) Eq. (11) Eq. (12).

$$k_{SE}(x, x') = \sigma^2 \exp\left(-\left(\frac{x - x'}{\lambda}\right)^2\right) \quad (10)$$

$$k_M(x, x') = \sigma^2 \frac{1}{\Gamma(a)2^a - 1} \left(2\sqrt{a}\frac{|x - x'|}{\lambda}\right) \mathbb{B}_a\left(2\sqrt{a}\frac{|x - x'|}{\lambda}\right) \quad (11)$$

$$k_E(x, x') = \sigma^2 \exp\left(-\frac{|x - x'|}{\lambda}\right) \quad (12)$$

The squared exponential kernel is a smooth, infinitely differentiable function. The hyperparameters σ^2 and λ are the output-scale amplitude (or the standard deviation) and the input time scale[23]. The exponential kernel with the same hyperparameters is less smooth and only once-differentiable. The Matérn kernel has an additional hyperparameter a renamed from the original paper to avoid confusion from the parameter ν in section 3.4.5. a is the degree of differentiability hyperparameter. Depending on its value, the Matérn kernel can become the squared exponential kernel ($a \rightarrow \text{inf}$) or the exponential kernel ($a = \frac{1}{2}$). This kernel can be used to balance the smoothness between the two extremes. Next to smoothness, the kernel function can be also chosen to capture periodicity in the data.

For the kernel, we chose a product of an exponential kernel and a periodic kernel with a squared exponential kernel base. This combination of a periodic and a static kernel is also called a *quasi-periodic* kernel.

$$k_{Qper}(x, x') = \sigma^2 \exp\left[-\frac{1}{2\lambda} \sin^2\left(\pi\left|\frac{x - x'}{T}\right|\right)\right] \quad (13)$$

This kernel is capable of capturing a change in the period through the static kernel part. By balancing the periodic kernel and the exponential kernel parameters during training, the model can prioritize the periodicity or the stationarity making it flexible. This choice was made based on some experiments where the time series of simulated covariances was plotted against the true (simulated) covariance. Other kernels like squared exponential error proved to be too smooth in many cases resulting in a nearly flat line that could not capture the dynamics of the covariance well.

Both Bayesian non-parametric methods described in the following sections will use the same quasi-periodic kernel. The hyperparameters will be set to the same values. The hyperparameters in question are σ^2 , λ and T . For the simulation data, we will use the values in Table 2. The hyperparameters for the ESM set were scaled with the expectation that depression progresses on the scale of weeks, not individual hours. While the symptoms might fluctuate, we are more interested in the long-term trends. As we saw in section 2.1 the course of depression can vary greatly from person to person. It is possible to recover within a year, have persistent symptoms longer than a year, or suffer from a cycle of remission and recurrence of symptoms. We will set the periodicity to have the scale of months to try to be applicable for both cases. These will be the initial values of the hyperparameters, however, they are all trainable (with exceptions that will be explained later).

hyperparameter	value simulation data	value ESM data
σ^2 (non-trainable)	1.0	1.0
λ	1.0	150
T	1.0	5000

Table 2: kernel hyperparameter values

Next to a kernel function, a Gaussian process is also determined by its mean function. Often, a zero mean is assumed. This mean has a consequence that the function will in many cases revert to a flat

horizontal line at the value 0. This is because the distance between data points determines how strongly the kernel function steers the Gaussian process. The Generalized Wishart process, further explained in section 3.4.5 uses the Gaussian processes to estimate the entries in the prior on the covariance. Since we do not want to assume either a positive or a negative relationship, it is reasonable to keep the zero mean function for the kernels belonging to the Gaussian processes considered. However, in both the multi-output Gaussian process method and the GWP, there are also Gaussian processes that try to fit the data output Y directly. We rescaled the symptom values to $[-3, 3]$, such that the zero mean also falls in the middle of the symptom values. Since we do not have enough expert knowledge on the topic of depression, we will keep it at those simple mean functions, to avoid making complicated assumptions about the data.

3.4.3 Multi-output Gaussian Processes with Coregionalisation

One can use several separate Gaussian processes to capture each output dimension independently of each other. However, if the output dimensions can be reasonably expected to be related, it makes more sense to also capture that relationship. Adding this relationship provides each individual dimension with additional information, which helps in case of missing data or noisy data. Observations from one dimension, influence the prediction in another dimension. There exist Gaussian Process methods that can do exactly that, here referred to as multi-output Gaussian processes (MOGPs). The method we will use in particular is based on a special kernel, called the coregionalization kernel. This kernel learns the free form covariance matrix over the outputs and uses it to aid the predictions. This output covariance will be later used to estimate the symptom network. The coregionalization kernel is formulated as follows:

$$k_{Coreg}(x_l, x_{l'}) = B(l, l') \quad (14)$$

where l and l' are two arbitrary output dimensions with $0 \leq l, l' \leq D$. B is a positive semi-definite matrix of the shape $D \times D$. The coregionalization kernel is often paired with a stationary covariance. In our case, we will use our quasi-periodic kernel Eq. (13). As mentioned before, there is a similarity here to the DCC-MGARCH model. We could draw a parallel between P_t and $k_{Coreg}(x_l, x_{l'})$; both have the function of combining independent processes with information about interrelatedness.

$$k(x_l, x_{l'}) = k_{Coreg}(x_l, x_{l'}) \cdot k_{Qper}(x_l, x_{l'}) \quad (15)$$

To ensure the positive definiteness of the matrix B , B is constructed from the Cholesky decomposition $B = LL^T$ with L being the lower triangular [24]. The coregionalization kernel has one hyperparameter: the rank. Rank has the function of reducing the number of parameters needed for B . A full rank B , meaning a B with $D \times D$ dimensions, requires us to estimate $D(D + 1)/2$ matrix entries. The rank is equal to the number of columns of the lower Cholesky decomposition L .

There also exists an alternative, more efficient solution to the problem, that being the multioutput Gaussian kernel class from GPflow [25] (see: [this tutorial notebook](#)). However, a problem with this method is that it does not have a readily available output covariance matrix. Although there exists a variant similar to coregionalization that mixes several uncorrelated Gaussian processes with each other through a mixing matrix W , W did not appear to be a good replacement for the coregionalization matrix. For this reason, we used the less efficient Coregionalisation method instead.

3.4.4 Coregionalization implementation

The implementation for this method was inspired by the notebook “A Simple case of coregionalization” [26] which showcases the use of an intrinsic coregionalization model for heterotropic data. Our selection of data points is largely homotropic, meaning that at each time point, all output variables have a value. There were only a handful of timepoints where this was not the case. The choice for this method was based on the easy access to the covariance between the output dimensions, represented by the coregionalisation matrix. This kernel class includes an `output.covariance()` function that retrieves said covariance. The example notebook used a VGP (variational Gaussian process) model. However, the example data

used there was significantly smaller than the data set used here. Using a large number of data points leads to enormous matrices which require a lot of RAM. When the model was trained on the entire dataset, the program crashed due to resource overflow. This forced us to use Stochastic VGP (SVGP), which uses inducing points and mini-batching to approximate the same problem at hand using fewer resources. The description in the tutorial for working with Coregionalisation [26] implies that the multioutput class itself is more optimized and therefore more efficient than the Coregionalisation kernel alone, without a mention of the optimization method. It remains unclear to what extent changing VGP to SVGP helps to mitigate the difference. An additional benefit to using SVGP is that we are also using this form of inference in our next model, the Generalized Wishart Process.

3.4.5 Generalized Wishart Processes

Alternatively, one could also try to model the relationship between the output dimensions using Gaussian processes instead of the other way around. This is the approach taken by the Generalized Wishart Process (GWP) method. In contrast to the coregionalization method, the relationship between the variables is captured dynamically with respect to the input dimension. The model estimates the covariance at each time point which is then used to sample Y predictions. Here we use this to model and predict the symptoms of MDD.

To understand GWPs, we first need to understand the Wishart distribution and its relation to the Gaussian distribution. The Wishart distribution describes the probability density function over positive definite matrices [9]. Therefore it can be easily used for describing covariance matrices Σ .

$$p(\Sigma | V, \nu) = \frac{|\Sigma|^{\nu-D-1}/2}{2^{\nu D/2} |V|^{\nu/2} \Gamma_D(\nu/2)} \exp\left(-\frac{1}{2} \text{tr}(V^{-1}\Sigma)\right) \quad (16)$$

$$\Gamma_D(\nu/2) = \pi^{D(D-1)/4} \prod_{j=1}^D \Gamma\left(\frac{\nu}{2} + \frac{(1-j)}{2}\right) \quad (17)$$

Here, V is a $D \times D$ positive definite scale matrix, ν stands for the degrees of freedom where $\nu \leq D$ and Γ_D is the multivariate generalization of the Gamma distribution and the chi-square distribution when ν is respectively real values or integer-valued. An important property of the GWP method is that the sum of outer products of multivariate Gaussian random variables follows the Wishart distribution.

$$\Sigma = \sum_{i=1}^{\nu} \mathbf{u}_i \mathbf{u}_i^{\top} \sim \mathcal{W}_D(V, \nu) \quad (18)$$

$\mathcal{W}_D(V, \nu)$ is a Wishart distribution with the parameters V and ν . \mathbf{u}_i is a Gaussian-distributed random variable with $\mathbf{u}_i \sim \mathcal{N}(\mathbf{0}, V)$. If we assume that the random variables \mathbf{u}_i are uncorrelated, we can also take the scale parameter out of the distribution. This allows us to describe the individual random variables as $\mathbf{u}_i \sim \mathcal{N}(\mathbf{0}, I)$. Then the Chomsky decomposition of the scale matrix called L can be used to arrive at the same Wishart distribution with $L\mathbf{u}_i \sim \mathcal{N}(\mathbf{0}, V)$, because $\mathbb{E}[L\hat{\mathbf{u}}_i \hat{\mathbf{u}}_i^{\top} L^{\top}] = LIL^{\top} = V$.

So far, the relationship described between applied to Gaussian random variables and the Wishart distribution of a single positive covariance matrix. However, we can extend this concept to processes as well. By replacing the Gaussian variables with Gaussian processes, we will obtain a Generalized Wishart Process; a collection of positive semi-definite random matrices that is in our case time indexed by time.

$$\Sigma(t) = \sum_{i=1}^{\nu} L \hat{\mathbf{u}}_i(t) \hat{\mathbf{u}}_i^{\top}(t) L^{\top} \quad (19)$$

In this equation, L is the lower Cholesky decomposition of the scale matrix V , with $LL^{\top} = V$, and $\hat{\mathbf{u}}_i(t) = (u_{i1}(t), \dots, u_{iD}(t))^{\top}$, where each individual Gaussian process $u_{id}(t)$ is defined as $u_{id}(t) \sim \mathcal{GP}(0, k)$. k is the kernel function such that $\text{cov}(u_{id}(t), u_{i'd'}(t')) = k(t, t') \delta_{ii'} \delta_{dd'}$. Here δ_{ij} is the Kronecker delta that ensures that the covariance is only determined by the rows and columns of $k(t, t')$ corresponding

to i , i' , d , and d' . We constrain the variance to follow $k(t, t) = 1$. This does not lead to a loss of generality as L can rescale the variance when needed. The Gaussian process functions at different times are jointly distributed with $(u_{id}(t_0), \dots, u_{id}(t_N))^T \sim \mathcal{N}(0, K)$. K is the $N \times N$ matrix with the elements $K_{ij} = k(t_i, t_j)$. There are νD Gaussian processes in total, because of the reach of the parameters i and d , namely: $i = 1, \dots, \nu$ and $d = 1, \dots, D$. The Generalized Wishart Process can also be summarised by Eq.(20) below.

$$\Sigma(t) \sim \mathcal{GWP}(V, \nu, k(t, t')) \quad (20)$$

To provide a more conceptual understanding of how each of the (hyper-)parameters involved influences the model, we will list them once again in Table 3.

parameter	interpretation
L (decomposition of the scale matrix)	the prior time-invariant expectation on the lower triangle of the Cholesky decomposition of Σ
ν (degrees of freedom)	determines the concentration around the prior on Σ . Smaller ν means a broader prior.
θ (hyperparameters of the kernel function)	describes the properties of the dynamics through the kernel function

Table 3: parameter interpretation of the GWP

For a more thorough explanation of the formulas, we refer to the paper by Wilson and Ghahramani [9].

GWPs offer us a way of describing a changing covariance matrix Σ using a kernel k that also enables us to use the properties of the change we want to capture like periodicity or smoothness. We then can use the GWP as a prior on the covariance matrix.

3.4.6 Training the model

But the question now remains, how do we arrive at the posterior Σ conditioned on the data? The parameters could be optimized through Markov chain Monte Carlo methods where the posterior is approximated through the sampling of the parameters [9][19]. We will use gradient-based variational inference instead, as it has a simple black box implementation and reduces the computational cost of inference [19].

This approach aims to maximize the evidence lower bound (ELBO) as defined later in Eq. (26). Its most important component is the log conditional likelihood since this is the non-black-box part that needs to be explicitly selected. Our choice fell on the likelihood function for the multivariate Gaussian distribution, but it could have some different distribution like the multivariate t-distribution. We chose the multivariate Gaussian likelihood because we wanted to be consistent about assuming the data to be Gaussian-like we also did for the DCC-GARCH implementation. The conditional log-likelihood is defined as follows:

$$p(Y | \boldsymbol{\mu}(t), \Sigma(t)) = \prod_{n=1}^N p(\mathbf{y}(t_n) | \boldsymbol{\mu}(t_n), \Sigma(t_n)) \quad (21)$$

Under the assumption that each of the variables $\mathbf{y}(t)$ is normally distributed, with

$$\mathbf{y}(t) \sim \mathcal{N}(\boldsymbol{\mu}(t), \Sigma(t)) \quad (22)$$

the likelihood can be further specified as:

$$p(Y | \boldsymbol{\mu}(t), \Sigma(t)) = \prod_{n=1}^N (2\pi)^{-D/2} |\Sigma(t_n)|^{-1/2} \exp \left[-\frac{1}{2} (\mathbf{y}(t_n) - \boldsymbol{\mu}(t_n))^\top \Sigma(t_n)^{-1} (\mathbf{y}(t_n) - \boldsymbol{\mu}(t_n)) \right] \quad (23)$$

with D being the number of output dimensions. The log-likelihood then becomes

$$\begin{aligned} \log(p(Y | \boldsymbol{\mu}(t), \Sigma(t))) &= \\ & \sum_{n=1}^N -\frac{D}{2} \log(2\pi) - \frac{1}{2} \log(|\Sigma(t_n)|) - \frac{1}{2} (\mathbf{y}(t_n) - \boldsymbol{\mu}(t_n))^\top \Sigma(t_n)^{-1} (\mathbf{y}(t_n) - \boldsymbol{\mu}(t_n)) = \\ & -\frac{ND}{2} \log(2\pi) \sum_{n=1}^N -\frac{1}{2} \log(|\Sigma(t_n)|) - \frac{1}{2} (\mathbf{y}(t_n) - \boldsymbol{\mu}(t_n))^\top \Sigma(t_n)^{-1} (\mathbf{y}(t_n) - \boldsymbol{\mu}(t_n)) \end{aligned} \quad (24)$$

To make the next step clearer, we will rewrite the log-likelihood by expanding Σ using from Eq. (19) and rewriting the entire equation for a single datapoint t . We will also collect the Gaussian processes at timepoint t into one matrix F_t with dimensions $D \times \nu$. The reformulation of log-likelihood at a single point then becomes

$$\begin{aligned} \log p(y_t | F_t) &= \\ & -\frac{D}{2} \log(2\pi) - \frac{1}{2} \log(|LF_t F_t^\top L^\top|) - \frac{1}{2} (\mathbf{y}(t) - \boldsymbol{\mu}(t))^\top (LF_t F_t^\top L^\top)^{-1} (\mathbf{y}(t) - \boldsymbol{\mu}(t)) \end{aligned} \quad (25)$$

The likelihood function is a part of the Evidence Lower Bound (ELBO) in Eq. (26) that is used in the variational inference. While training the model, we try to maximize the right-hand side of the equation.

$$\log p(Y) \leq \sum_{n=1}^N \mathbb{E}_{q(F_n)} [\log p(Y_n | F_n)] - \sum_{d=1}^D \sum_{i=1}^{\nu} KL [q(U_{d,i}) || p(U_{d,i})] \quad (26)$$

This implementation of the GWP uses variational inference together with M inducing points Z . The evaluations of the Gaussian processes at these points Z will here be indicated with $U_{m,d,i} = u_{d,i}(Z_m)$. The inducing points are within the same scope as the data X . Using fewer inducing points than datapoints ($M < N$) reduces the computational complexity of learning. $q()$ is the surrogate probability distribution that tries to approximate the true posterior. KL is the Kullback-Leiber divergence that can be evaluated analytically to construct the gradient. $q(F_n)$ is not conjugate to the likelihood $\log P(Y_n | F_n)$ so the gradient is obtained through differentiating through random samples from q in a kind of Monte Carlo approximation.

The end product consists of the gradients for the inducing points Z , the kernel parameters θ , the scale matrix V , and the variational parameters that define the distribution q . The variational inference is implemented using the Sparse Variational Gaussian Process (SVGP) class from the GPflow package[25]. This also includes the addition of white noise γ of shape $D \times D$ to Σ in the log-likelihood function. The white noise is a matrix with non-zero values only on the diagonal. The white noise is placed there to avoid extremely high variance of the Monte Carlo gradient approximation part of the model, which was observed by Wilk and Heaukulani [19] to prevent the parameters from moving towards their desired value space.

The GWP code in this project comes largely from the Bayesian Network Regression package (BANNER) that was built on top of the GPflow package. The entire package can be found on Github [8]. BANNER is essentially the implementation of the paper by Heaukulani and Wilk [19] using GPflow 2.0. Heaukulani and Wilk focused on modeling the covariance only. The mean (μ , not the mean function) was assumed to be a zero-mean for all the output dimensions. When we try to model the symptoms of depression, it is not only essential to model the covariance, but also the estimated mean value of the symptoms. Otherwise, all y_t samples would be taken from the distribution $\mathcal{N}(0, \Sigma_t)$, even when

all outputs around that time point are clustered around other value for example -2 and sampling from $\mathcal{N}(-\mathbf{2}, \Sigma_t)$ would be more reasonable. For this reason, BANNER has been extended to estimate the mean as well as the covariance over time.

The assumption of using a zero $\mu(t)$ was reflected in the likelihood formulation and the sampling formulation. The more general likelihood formulation as defined by Wilson and Ghahramani is reflected by Eq. (24). The BANNER version was using $y(t_n)$ instead of $\mathbf{y}(t_n) - \boldsymbol{\mu}(t_n)$ which was easily adjusted by adding the μ term. In the same way, the sampling of $y(t)$ was changed back from $\mathbf{y}(t) \sim \mathcal{N}(\mathbf{0}, \Sigma(t))$ to $\mathbf{y}(t) \sim \mathcal{N}(\boldsymbol{\mu}(t), \Sigma(t))$

Following a suggestion by Wilson and Ghahramani, $\mu(t)$ was added as a Gaussian process with three possible variations, which we call respectively “fully dependent”, “shared”, and “independent”:

$$\boldsymbol{\mu}(t) = \sum_{i=1}^{\nu} L\mathbf{u}_i(t) \quad (27)$$

$$\boldsymbol{\mu}(t) = \sum_{i=1}^{\nu} L\mathbf{u}_i(t) + \mathbf{u}_{\nu+1}(t) \quad (28)$$

$$\boldsymbol{\mu}(t) = \mathbf{u}_{\nu+1}(t) \quad (29)$$

Each of the variations has different implications. Eq. (27) is a mean function that is equal to the variance. It implies that the mean gets higher as the prediction becomes more uncertain. This could be a good choice if we expect the magnitude of the symptoms to increase as their volatility increases.

Eq. (28) still implies a relationship between volatility and the mean, however, it is no longer the sole source for the mean. μ gets its own degree of freedom $\nu + 1$ which allows the mean to behave differently than Σ . Finally, Eq. (29) separates the mean from the volatility. This is a good choice if we don’t expect Σ and μ to be positively correlated.

The zero mean approach is also still possible, by simply trying to enter “zero” as the mean estimation method. This results in $\boldsymbol{\mu}(t) = \mathbf{0}$ for all time points t

In our implementation we used the “independent” mean estimation from Eq. (29). This option was opted for, because the strength of each individual variable like positive affect or mental unrest, may correlate differently with the elements of the covariance matrix. In fact, one reason to use the network approach in psychopathology research is to investigate those relations. Therefore, we avoid assuming that the value of the output and the strength of the connections are correlated from the start, such that the distinction between symptom outcome and the relationship strength can be assessed outside of learning.

Other implementation choices include the kernel choice, training parameters, and inducing points.

We used the quasi-periodic kernel described in Section 3.4.2. Each of the $\nu D + D$ Gaussian processes has the same kernel type, however, the parameters for each kernel are learned separately. This is implemented through the GPflow SeparateIndependentMok kernel. We chose the number of inducing points M to be 20% of the total number of data points N . The inducing points are trainable and shared for each output dimension. They are contained in a SharedIndependentInducingVariable object from GPflow.

The training hyperparameters were also chosen after some experimentation. When looking at the ELBO values, the model should be sufficiently trained after approximately 5000 iterations. That is where the ELBO line seems to be stagnating. However, the ELBO value appeared to be a bad indicator of the number of iterations needed, because the model only started performing well after approximately 80,000 iterations. Other training hyperparameters included learning rate, and batch size, set to 0.01 and 100 respectively.

As of today, the GPflow 2.0 package that our implementation of the GWP is based on does not support online learning. Some methods could be incorporated in the future to accommodate this feature. A fitting method would for instance be Streaming Sparse Gaussian Process Approximations [27]. This method is based on collapsed variational free-energy and Power-EP. Expanding BANNER further to accommodate online learning was left for future research.

3.4.7 BANNER and scalability

A shortcoming of Gaussian Processes is their scalability to large datasets. The greatest caveat is the inversion of the covariance matrix which becomes computationally heavy when N is large.

We try to solve this problem through Stochastic Variational Inference. In essence, this method estimates the GP posterior using a GP conditioned on a smaller set of inducing points. Those inducing points are supposed to summarise the true dataset. (https://gpflow.github.io/GPflow/development/notebooks/advanced/gps_for_big_data.html) In addition to that, it also uses mini-batching. Mini-batching improves scalability since the algorithm always trains on a subset of the dataset. Due to smaller portions of data being processed at each iteration, costly operations like matrix inversion become more efficient.

3.5 Data processing

The ESM dataset consists of 1476 moment measurements. The daily questionnaire contained 12 mood variables, 4 mental unrest variables, 4 self-esteem variables, 9 social setting variables, 8 physical symptom variables, 6 activity variables, and 13 event variables, for a total of 50. This excludes the questions that were asked only once a day or weekly. Those are too many variables for the models to remain interpretable. Furthermore, Gaussian models do not scale well in a too large output dimensionality. Therefore, we made a selection of variables to use in our project. An important fact to keep in mind is that this introduces new assumptions into the model. For instance, the grouped variables are statically linked to each other throughout the entire duration of the experiment.

We decided to use the same set of variables as the author of the original ESM dataset paper [16] since it also contained symptom network models that we will be able to compare. Said paper categorized a selection of the variables into “positive affect”, “negative affect”, “mental unrest”, with the addition of the items “worrying”, and “suspicious” to capture the current cognitive context and subclinical psychotic experiences respectively. We will group the existing variables as follows:

- **negative affect:** down, lonely, guilty, doubtful, anxious
- **positive affect:** enthusiastic, strong, cheerful, satisfied, relaxed
- **mental unrest:** irritated, agitated
- **subclinical psychotic experiences :** suspicious
- **cognitive context:** worry

The values of the new categories were computed as an average of all the variables they are composed of. The scale of each category was normalized first to fit the range [-3,3]. This sets the output scale mean to zero, which allows us to keep the standard zero-mean function for the Gaussian process methods.

For a non-disclosed reason, the affect symptom “mood down” was excluded from the analysis conducted by Wichers et al [16]. This symptom seems to be at the core of depression, as the DSM 5 states that “Five (or more) of the following symptoms have been present during the same 2-week period and represent a change from previous functioning; at least one of the symptoms is either (1) depressed mood or (2) loss of interest or pleasure”. The symptom was included in our negative affect category for this reason.

We used the meta-variable indicating the day number together with the time of the beep signal to create an input variable expressed in hours since the start of the experiment. This variable X ranges from 33 to

5758. We also dropped the rows that contain missing values. This reduces the dataset size to $N = 1473$. GWP and MOGP did not require any further major adjustments, except for some reshaping to fit the input format. Namely, our BANNER implementation of the GWP requires the input to be shaped as $(D \times N)$ where each column is the same. MOGP requires the input data to be reshaped such that each input x consists of the value x and the label l . The same thing holds for the output Y . The final form of the data for MOGP is $X : (ND \times 2)$ and $Y : (ND \times 2)$.

MGARCH, on the other hand, requires an additional extra step that goes a bit further. MGARCH requires regularly spaced data to work. Although there exist GARCH models designed to handle irregularly spaced data, like ACD-GARCH [28] [29], to the best of our knowledge there are no readily applicable models that also can handle multivariate data. For this reason, the data needed to be resampled accordingly. There are many ways of achieving this, often involving some form of interpolation. One method is to impose a ticking frequency and take the value that is the closest in time. Another one is a form of linear interpolation where we take the weighted mean between the two neighboring values where the value the closest in time gets the most weight. We are using a combination of those, where the data gets grouped into blocks of 4 hours. When there are one or multiple values within that block, the first value will be picked. Otherwise, the value will be interpolated using its neighbors. The result is the weighted mean between the two closest values.

3.6 Data simulations

The real (dynamic) covariance of the dataset is unknown. We will simulate the data from a known precision matrix instead. This can be divided into two steps: creating a temporally changing covariance, and sampling new data using that covariance.

For the first step, we will start by creating a precision matrix. As mentioned before in Section 3.2, this is how our symptom networks are represented. Starting from a precision matrix helps us be more clear about the networks and the changes in the relations that we are simulating.

The simulated precision matrices have the shape of $D \times D$ with $D = 3$ and they are sampled over a period of $[0, 6.0]$. There are $N = 500$ samples on regular intervals. The implementation includes three ways of generating precision matrices. The first way is periodic, where the time series is based on two alternating precision matrices. The values on the diagonal are set to 1, such that the correlation of a variable with itself would be 1. The off-diagonal values chosen were $\{0.5, -0.1, 0.1\}$ and $\{-0.3, -0.6, 0.8\}$ for the first and the second matrix respectively, such that $K_{1,0,1} = 0.5$, $K_{1,0,2} = -1.1$, $K_{1,1,2} = -0.1$, and the upper and lower triangle of the matrix are the same. The precision matrices alternate with a period of 0.7. The second generation method depends on linearly changing the precision over time. The same two matrices are now used as the starting point and the end point of a linearly changing network. At each time step t the precision matrix is computed as $K_t = K_1 + (t - t_{start}) \cdot step$, where $step = (K_N - K_1)/(N - 1)$. Finally, the last method is just an unchanging precision matrix equal to the first matrix of the periodic method.

The precision matrix then gets transformed back into a covariance matrix through inversion. The covariance matrix is checked for positive definiteness by attempting to perform Cholesky decomposition on it. This results in an error when the property is not met.

Next the covariance is used to sample data points from a multivariate normal distribution, following $y \sim \mathcal{N}(\mu, \sigma^2)$. We use different means for each dimension to make the simulation a bit more complex. We use

- μ_0 : straight line from $(0, 0)$ to $(T_{max}, 4)$
- μ_1 : cosine function $\cos(x)$
- μ_2 : constant mean at $y = -2$

We sample N D -dimensional data points during the interval $0.00 - T_{max}$. Each run of the simulation produces a different set of data points. Due to chance, each run may differ in the representability of the underlying covariance, making it harder or perhaps even impossible to learn accurately. This underlines the difficulty of the problem at hand, where the data is the only thing we have and where we need to assume that it is representative enough for the covariance. We repeat the sampling of each dataset five times. Ideally, we would do it at least ten times, however, training each model ten times was not feasible due to time constraints.

3.7 Experiments

For our first experiment, we want to predict the covariance over time based on the simulated data. The data conveys some characteristic trends that can be expected in time series (periodicity, linear change, stable covariance). There are countless possibilities for simulating a dataset, but here we experiment only with the aforementioned three kinds. Since the data is randomly sampled from the given parameters Σ and μ , the representativeness of the data can vary from run to run. We sample the data five times from each kind of partial correlation and save them. We use the same simulated data consistently for all the models in order to facilitate a fair comparison.

All the simulated datasets had $N = 500$ and $D = 3$. The estimated covariance will be plotted on top of the ground truth covariance. The log-likelihood of the data, the mean squared distance of Σ , and the correlation between the ground truth and the estimated Σ will also be computed for each model. Those values will be averaged over all samples and time points to obtain a mean metric value.

We also want to assess the quality of our proxy method for estimating the quality of the models on the real data. We will compute the output predictions for each model using the learned parameters from the previous experiment and compute the correlation between our Σ related metrics and the predicted output Y_{pred} related metrics. We expect to find a negative correlation between the log-likelihood and the mean squared error of Σ since we expect the data to be more likely under a Σ_{pred} that is closer to the real Σ . A positive correlation found between one of the other proxies (Y MSE or Y correlation) and the Σ MSE may also indicate their adequacy.

With the actual ESM dataset, we can only use log-likelihood, the mean squared error of the output, and the correlation of the output. We will collect those three metrics from all models to compare their predictive capabilities. All three metrics will be computed on out-sample data. Whether that helps us determine which method is best for the modeling of the covariance depends on the quality of the proxy.

Finally, we will plot the network progression resulting from the GWP method as a proof of concept. We will compare that graph to the results from the paper by Wichers [16]. The network will be based on the partial correlations (Eq. (3)):

3.8 Evaluation

We want to know whether GWP is an interesting modeling method for this particular problem. First, we want to measure its predictive power to see whether it can capture the behavior of the data. Since the data is structured in a time series, we cannot simply use the regular k-fold cross-validation method for dividing the dataset into train and test data. The data is temporally dependent and the order of data points cannot be mixed. We use cross-validation for time series, where the data order is not randomized. Instead, the data is divided into a number of folds. First, it is trained on the first fold and tested on the second, then it is trained on the first and second, and tested on the third etcetera. The test predictions are accumulated throughout the entire training process. With the simulation data of size $N = 500$, we use the last $N_{test} = 40$ datapoints for testing. The prediction horizon is set to 5, meaning that after each training fold, the model will predict the next 5 data points. Those 5 data points will be included in the training set in the next iteration. The process is repeated until all the test data has been predicted. This is done with all of the models, although with small differences. None of our implementations sup-

ports online learning, meaning that each new fold needs to be retrained entirely. It is not a problem for MGARCH which is comparatively extremely fast. For this reason, the model can be retrained from scratch with each newly added testing window. GWP and MOGP are too slow to feasibly retrain from scratch every time. To learn sufficiently well, both models require a number of iterations close to 100,000. Given that they would need to be trained $3 * 5 * 8 = 120$ times, it would most probably take weeks to run using our current resources. This is why we opted for initially training the model for 65,000 iterations and then training the same model 5000 iterations more with the addition of each new subset of the data. This choice was made from the assumption of applicability of incremental learning to Gaussian processes. This assumption was made based on our observation of the importance of the (hyper)parameter values for the model. We reasoned that training the model on a large portion of the time series and with a considerable number of iterations would lead to reasonable initial values for each consecutively added data subset. In consequence, the subsequent iterations would not need as much training, being already steered in the right direction before optimization. Finally, the accumulated predictions are used to calculate the mean squared error and the cross-correlation between the predictions and the real test data outputs as described in the previous section.

4 Results

4.1 Simulation data study

In Figures 1,2,3 we show the average covariance predictions for each model over time. The red lines are based on all the seen data and the blue lines are the future predictions. The MOGP model is changing in the predictions, because every 5 samples are based on another set of data, leading to slight changes in the covariance. For the periodic data, both the MGARCH and the GWP recognize the pattern. MGARCH is much noisier, while GWP is smooth. Neither model average reaches the correct amplitude.

In the linearly changing partial correlation case, MGARCH appears to be more capable of catching on to the new trend and following the changing line. GWP has trouble adjusting. For the MOGP, the covariance prediction is not following the line very well, even in the flat parts.

Finally, in the case of a constant static partial correlation, MGARCH and GWP seem to follow the static covariance reasonably well, although the uncertainty for GWP is much higher.

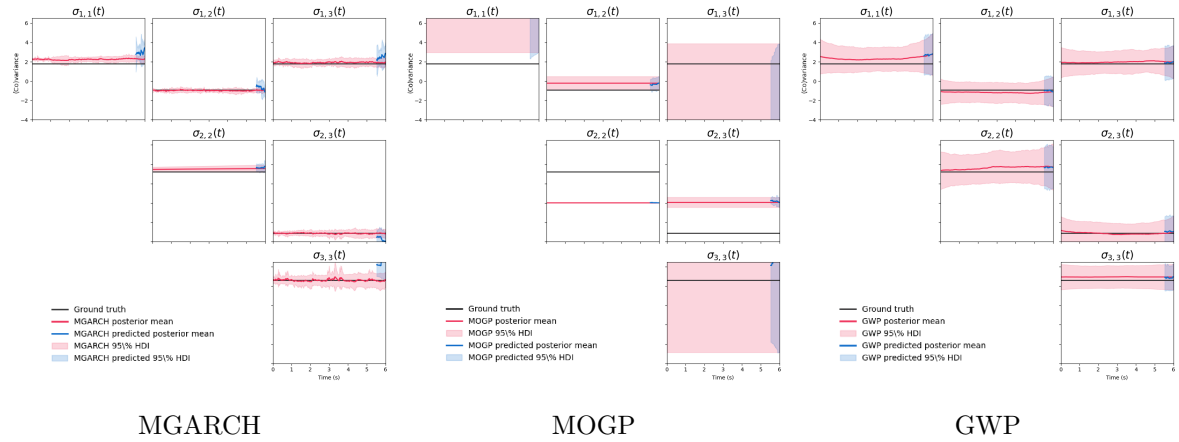


Figure 3: The estimated covariance matrices plotted against the true covariance derived from the **static** precision matrix

Table 4 shows the average metric values for all models in all simulation cases. MGARCH is the best scoring model in terms of likelihood. MOGP is consistently the best in terms of the MSE of the output prediction Y . The correlations between the output Y and the prediction of Y are negligible for all models. MOGP is scoring the worst overall for the direct metrics of the covariance fit. MGARCH seems to be overall the best based on the direct covariance-based metrics.

	MGARCH	GWP	MOGP
periodic (period = 0.7)			
log-likelihood	-5.6198	-5.84380	-6.61973
MSE (y vs ypred)	6.84252	6.09027	4.30962
Correlation (y vs ypred)	-0.01602	0.00277	-0.01569
MSE (sigma vs sigmapred)	1.32704	1.55071	20.2267
Correlation (sigma vs sigmapred)	0.09190	0.09914	-0.02355
linearly changing			
log-likelihood	-5.53252	-5.87194	-6.92721
MSE (y vs ypred)	6.33190	5.54903	4.45330
Correlation (y vs ypred)	-0.00722	-0.00262	-0.01266
MSE (sigma vs sigma pred)	0.43942	1.45204	7.50182
Correlation (sigma vs sigmapred)	0.46634	0.34004	-0.01517
stable			
log-likelihood	-5.59766	-5.83025	-6.74915
MSE (y vs ypred)	8.28842	7.56091	4.87079
Correlation (y vs ypred)	-0.013184	-0.00566	-0.06159
MSE (sigma vs sigmapred)	1.01660	0.90852	32.21070
Correlation(sigma vs sigmapred)	None	None	None

Table 4: A summary of the metrics on the simulation data. The white rows indicate the proxy metrics. The gray rows indicate the metrics directly measuring the fitness of the Σ prediction. The bold value is the best score of the row.

Table 5 displays the values of the correlations between the proxies and the MSE between the covariance ground truth and prediction. For log-likelihood, a negative correlation was desirable because we want to maximize the log-likelihood, and we want to minimize the MSE. For the Y output MSE, the desirable

correlation is positive, since we want to minimize both kinds of MSE. Overall the correlations were weak. None were higher than 0.2. The correlations were also inconsistent across the models.

	MGARCH	GWP	MOGP
periodic			
log-likelihood	0.17530	0.01495	-0.0027
MSE	-0.08314	0.03515	0.18974
linear changing			
log-likelihood	-0.17319	0.014534	-0.07236
MSE	0.06859	0.040582	0.08091
stable			
log-likelihood	0.03708	0.01142	-0.01071
MSE	-0.02292	0.00816	0.08783

Table 5: This table describes the correlation between the MSE of the covariance and two of the proxies (log-likelihood and the MSE of Y).

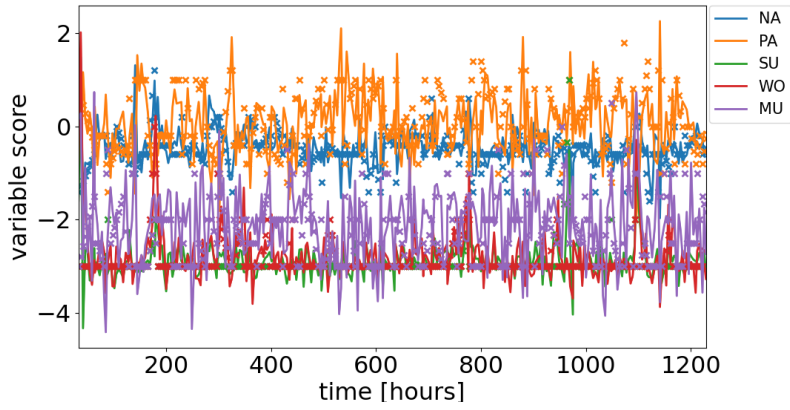
4.2 ESM data study

Table 6 shows the final results based on the ESM data. MGARCH had the best log-likelihood score and correlation score. MGARCH’s log-likelihood also had the lowest variance. MOGP had the best MSE score with the lowest variance. MGARCH’s score for the correlation between the output and the prediction of Y was by far the best with still a low variance.

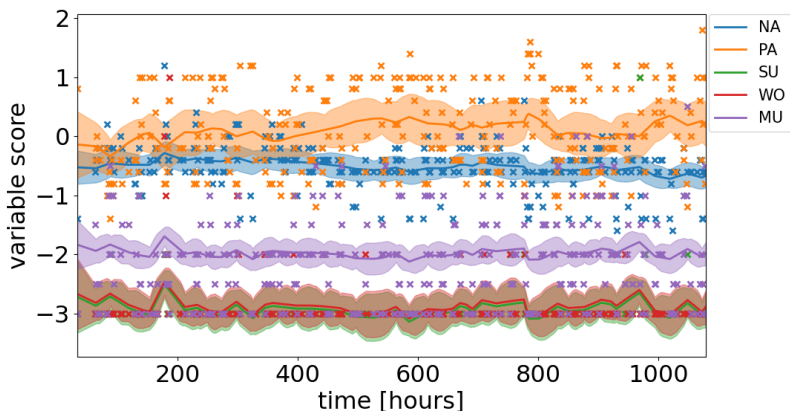
	Log-likelihood	MSE	Correlation
MGARCH	-3.8245 ($\sigma^2=0.7639$)	2.1005 ($\sigma^2=17.7328$)	0.3950 ($\sigma^2=0.0375$)
MOGP	-8.0780 ($\sigma^2=18.9026$)	1.3745 ($\sigma^2=3.8463$)	-0.0886 ($\sigma^2=0.0691$)
GWP	-6.9938 ($\sigma^2=6.6132$)	2.4051 ($\sigma^2=17.3653$)	-0.0046 ($\sigma^2=0.03054$)

Table 6: Summarising performance metrics based on the forecasted data. The best-scoring metric is **bold**. We indicated the variance in the brackets behind the value

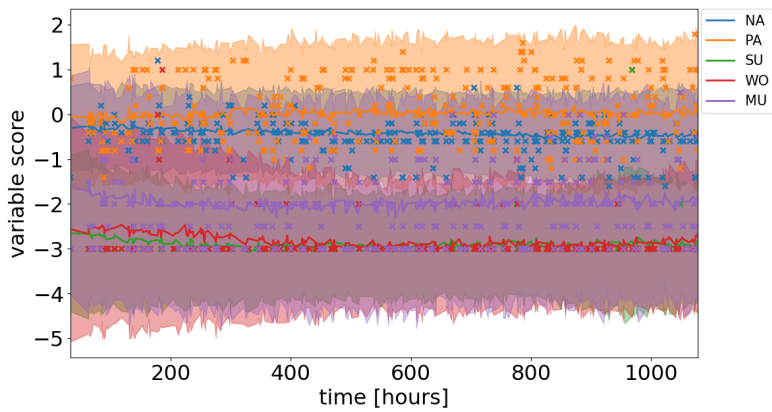
Fig. 4 shows the (historical) Y predictions for the first 300 data points. The abbreviations in the legend stand for “negative affect”, “positive affect”, “suspicious”, “worry”, and “mental unrest” respectively. The 300th datapoint translates to the 1089th hour from the start of the experiment in the original set and the 1232nd hour in the resampled dataset. We took a subset of the data for readability purposes. In terms of smoothness, the models can be arranged in order MOGP-GWP-MGARCH from best to worst. MOGP and MGARCH have much tighter 95% HDI bounds than the GWP. MGARCH’s prediction looks like it is fitting directly to the known data points.



(a) MGARCH



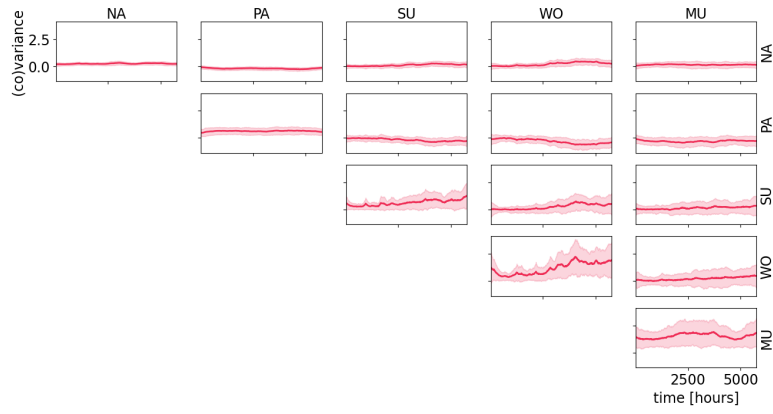
(b) MOGP



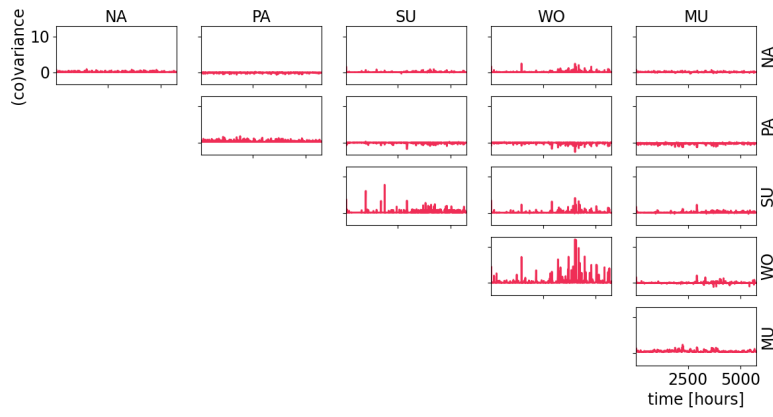
(c) GWP

Figure 4: Time series model predictions of the first 300 datapoints

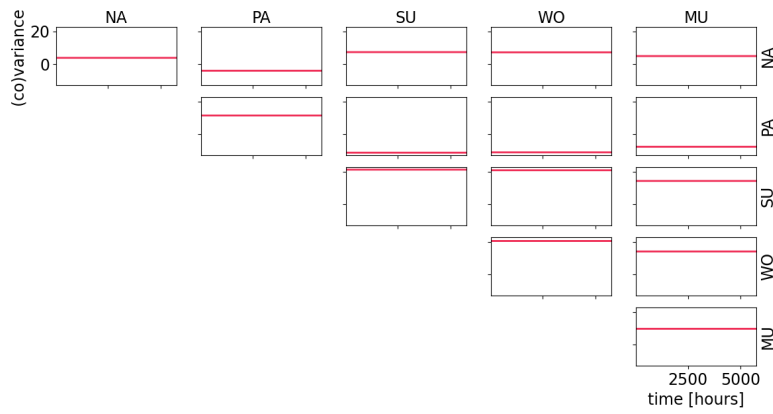
Figs. 5,6 display the final covariance and partial correlation estimations. GWP shows trends that are easy to decipher and interpret, thanks to the relative smoothness. While MGARCH might have scored higher on the likelihood, we find it harder to see trends in the partial correlation grid in Fig. 6b. The partial correlations produced by the MOGP do not seem to align with the results from other models. For instance, the partial correlation between positive affect and negative affect appears to be negative at all times. Common knowledge also supports the negative partial correlation between the two. Nevertheless, the trend seems to be around zero for MOGP. In fact, MOGP attributes a near zero partial correlation for almost all combinations, except for (worry, mental unrest) and (suspicious, mental unrest).



(a) GWP

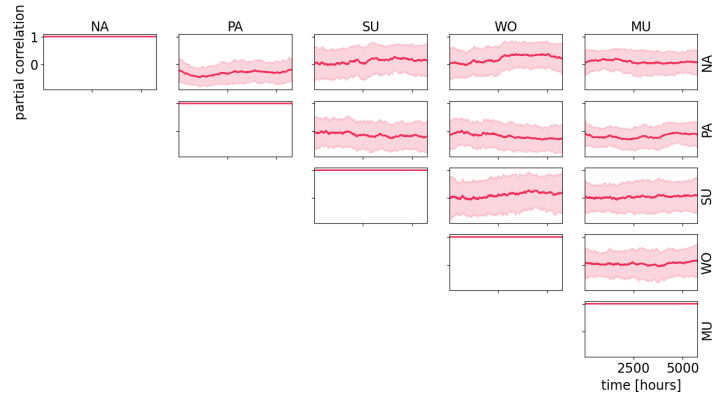


(b) MGARCH

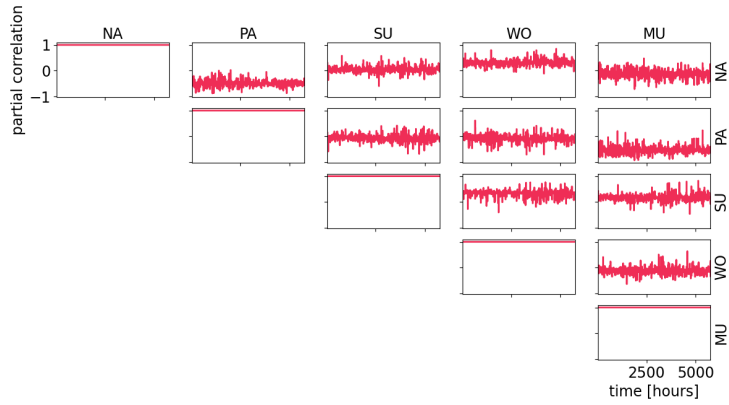


(c) MOGP

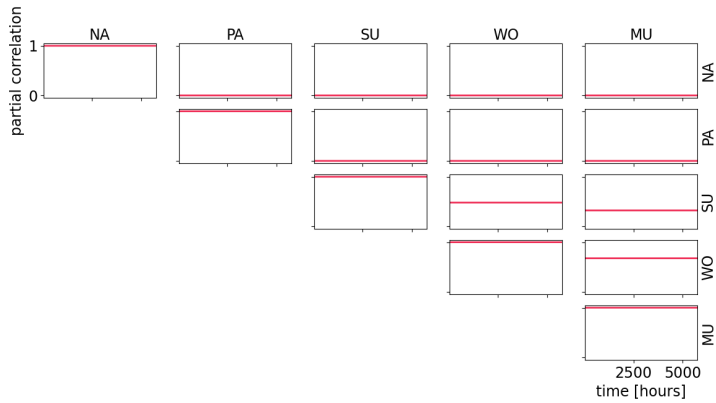
Figure 5: Covariance prediction of the respective models



(a) GWP



(b) MGARCH



(c) MOGP

Figure 6: Partial correlation prediction of the respective models

4.3 Network results

Here we will plot the resulting network structures for the GWP model and compare them to the network structures from the paper that our ESM data originates from [16] in Fig. 7. To prevent cluttered networks, we omitted the self-loops, which were all set to 1, and we set the threshold for showing edges to

0.03. We chose this threshold since this is a very weak correlation and correlations lower than that are usually seen as extremely weak. In our plots, red values are negative correlations and black edges are positive correlations. The thickness of the edges in our plots indicates the strength of the connection.

An important difference between our models in Figures 8, 9 and the reference model in Fig. 7 provided by Wichers et al [16], is that their model is a directed graph. The values in the graph reflect the Beta coefficients, instead of correlations. Therefore, we cannot directly compare the graphs' values, since they are picturing slightly different scenarios. Nevertheless, Beta coefficients are related to correlations (see [this explanation](#)) and we can try to derive some trends. The first thing that can be observed in the reference figure Fig. 7 is that one of the strongest connections in the GWP and the MGARCH models, (positive affection - negative affection) is not present in the first two phases of the experiment and even in the last two, it is relatively weak. Conceptually, one would also expect this connection to be strong, as it represents opposite mood types. In all models, the number of connections, and the strength of connections seem to increase as time goes on with the dose of the antidepressant (AD) being lowered. Another interesting difference between the GWP model and the reference is that the mental unrest - negative affect connection has an opposite relationship in the last phase of the experiment. The edge is positive in the GWP model and negative in the reference model. The same edge is negative in the MGARCH model. The connections in the MGARCH graph are in general stronger than the connection in the GWP graph. In many cases, they overlap at least in the positive/negative edge distinction. This is not the case in the aforementioned negative affect - mental unrest edge, and the mental unrest - worry edge in the last phase in the image.

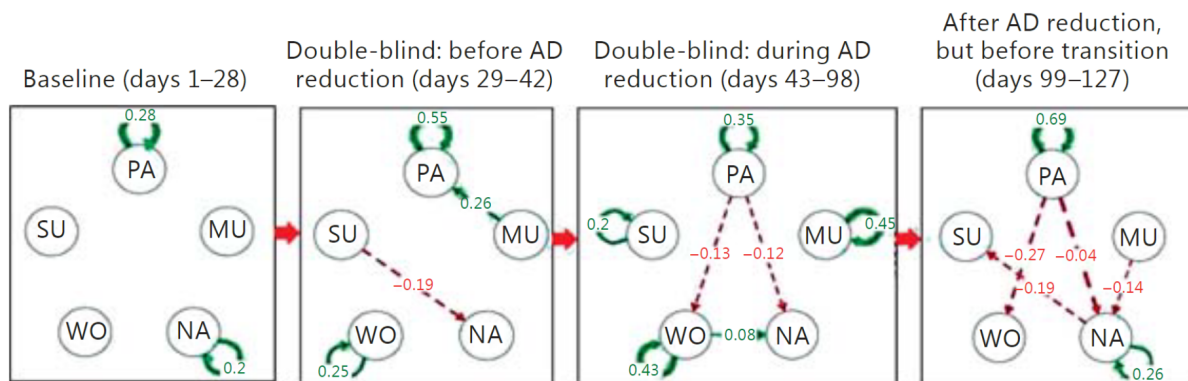


Figure 7: AD = antidepressant, PA = positive affect, NA =negative affect, MU = mental unrest, SU =suspicious, WO =worry. Spatial correlation networks based on the ESM data as plotted by Wichers[16]. The network edges represent the beta coefficients.

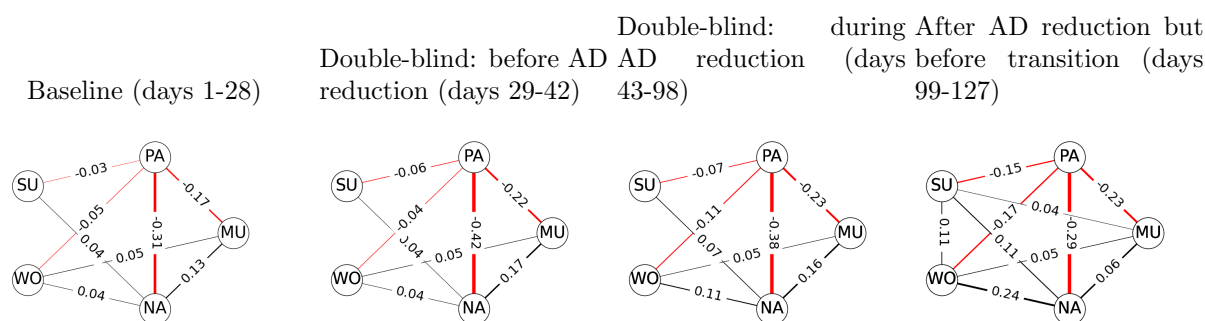


Figure 8: Partial correlation networks modeled by the GWP.

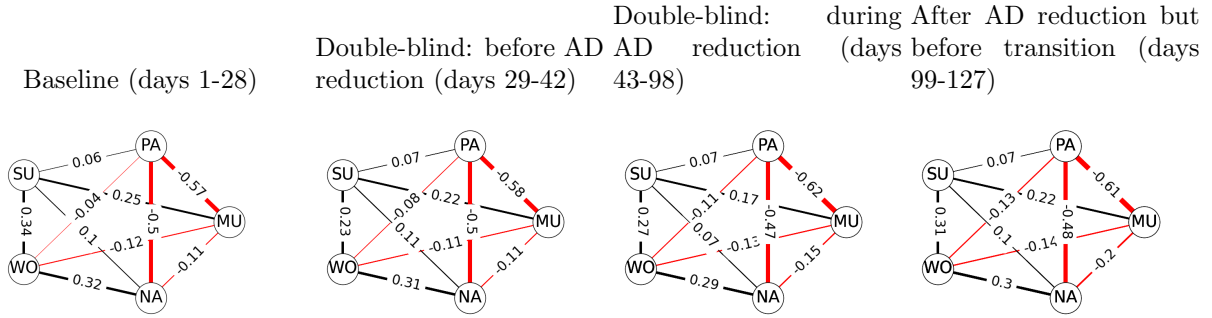


Figure 9: Partial correlation networks modeled by MGARCH.

4.4 Performance observations

Some of the things observed during the experiments are the training times and requirements of each model. MGARCH was by far the fastest and could be trained on the ESM dataset within around 15 seconds using our machine (a laptop with a Intel(R) Core(TM) i7-7700HQ CPU, 16GB RAM, and Intel(R) HD Graphics 630 GPU). MOGP and GWP required a lot of training iterations before they could perform sufficiently well. While MGARCH’s run time was on the scale of seconds even without the GPU, MOGP and GWP had often to run for hours up to days. The use of GPU for those models made the training faster because much of the time is lost due to costly matrix operations (inversion). The MOGP model originally used a VGP (variational GP) instead of an SVGP. We adapted it to the SVGP because the VGP does not support mini-batching. When the model was training on the entire ESM dataset without mini-batching, it would have to hold the entire kernel of over 7000 by 7000 entries in memory which caused a RAM overflow and crashed the program. Furthermore, MOGP and GWP were strongly dependent on the initialization of the kernel hyperparameters, while MGARCH worked without any tweaking of hyperparameters needed.

5 Conclusion

One of the questions of this thesis is whether the observed symptoms of MDD could be better predicted when the dynamics of their interactions were explicitly captured. As we saw in the results, MOGP, the model that only learned the static covariance systematically had a better MSE score when predicting the output, while having the worst scores for covariance approximation. It is still possible that dynamically capturing those relationships can lead to better output predictions. However, that was not the case when we compare the three models in this project.

The second question was more general and focused on the actual modeling part. The question was: “Can we capture the properties of MDD in terms of covariance functions and mean functions in a nonparametric Bayesian model?”. We saw that we could formulate a Generalized Wishart Process to tackle the problem. We used a flexible yet simple kernel and a mean function. There was little information on the temporal dynamics of the MDD that we could capture in model assumptions, therefore we tried to use a kernel that could capture periodicity as well as nonperiodic dynamics. The model was able to roughly capture the covariance dynamics in simulated data, although it performed worse than the parametric baseline. The baseline also performed the best on the ESM dataset based on the proxies. So while the answer is: yes, it is possible to model depression (symptom relations) using a nonparametric Bayesian model, so far it does not perform better than the baseline based on the metrics. However, it had the nice property of smoothness which made the results easier to interpret.

6 Discussion

The graphical results in Fig. 6 were a promising glimpse into the interpretability and the informativeness of GWP models. Thanks to the smoothness of the GWP, it is easy to follow the progression of the connection strengths. The uncertainty bounds give a good impression of the reliability of the resulting models. While the GWP underperformed when compared to our nonparametric baseline, we did not encounter any indications of the model being conceptually unfit to target the problem of estimating the dynamic covariance of ESM data. The simulation data results have shown that the GWP is capable of capturing dynamic covariance reasonably well.

We tried not to make too many MDD-related conclusions based on our results. The focus of this project is on the methodology, not the outcomes. Nevertheless, the incentive for this project was the idea that it would be useful to model symptom networks dynamically, therefore it is interesting to reconsider this in the light of our results. When inspecting the individual edge progressions in Fig. 6 and the plotted networks in Fig. 8 and Fig. 9, it becomes clear that the (averaged) MGARCH output does not change a lot. The GWP on the other hand shows some interesting shifts, most notable of which being the worry-negative affect connection, which changes from approximately 0.05 to 0.25 later on in the experiment. Other relatively dynamic edges were worry-suspicious and worry-positive affect. Our GWP results seem to be in favor of modeling the symptom networks dynamically. This was also the case for the reference networks in Fig. 7.

Of course, the results are nongeneralizable due to the sample size of only one person. The real MDD research is left to the psychiatry researchers. In our opinion, GWP has the potential of becoming a valuable modeling tool for network dynamics, thanks to the aforementioned interpretability. However, we would not recommend this method to be handed over from the machine learning community to psychiatry research just yet. The GWP has many moving parts that need to be chosen carefully and require a solid understanding of the Gaussian processes and related topics. We need to learn why it did underperform in this case, how it can be prevented in the future, and how we can make it less sensitive to its starting hyperparameter values.

As we saw in the Conclusion (Section 5) and the sections before that, our proposed model did not perform better than the baseline. One possible reason is that our naive training schedule did not allow the model to train sufficiently well. As for now, the GWP implementation does not support online learning. It is very well possible that training the model for 65,000 iterations and then adding 5,000 iterations to that for each 5 new test points was not working as hoped. This choice was made at risk because due to time limitations, other alternatives like retraining the model entirely each time with 100,000 iterations or trying to incorporate online learning on our own were not feasible. The required number of iterations was tested on a smaller dataset ($N = 500, D = 3$) where 80,000 worked well, so it is also very well possible that the GWP was just not sufficiently trained.

Another possible reason is of course that the implementational choices made on the way were suboptimal for the problem at hand. Due to time constraints, the real experiments were conducted only with one type of kernel. If the time allowed it, we would have repeated all experiments using other kernels like the simple exponential, and square exponential kernels on their own. Even in the case where we knew all the data properties, the GWP performed slightly worse than the MGARCH. In the case of simulation data, the hyperparameters of the kernels were set to be close to their optimal values, yet the model did still not perform better than its parametric alternative.

Another issue that needs to be addressed, is the unreliability of the proxies. The correlations between the proxies and the Σ predictions as measured by us were not satisfactory. However, log-likelihood has been used as such proxy before by other authors [9]. Furthermore, it is conceptually a logical proxy to use. It describes how likely the data is given our parameters. Logically, parameters Σ and μ will probably be better at approximating the “real” data distribution if they are near their correct values. At the same time, the likelihood does not require the true Σ to be known, which is a problem with the

distance-based metrics. Although the connection between the y prediction accuracy and the Σ prediction accuracy was also not shown numerically, the output prediction accuracy is nevertheless interesting. To our knowledge, there are no better-fitting proxies than the ones used in this project. For this reason, we still decided to keep the results for these metrics in the thesis, although we want to emphasize that one needs to be careful when using them to conclude anything about how good a model is at capturing the covariance.

In a follow-up to this research, we would like to address the question of how we can make the GWP more robust. We would like to know whether it is a matter of improving the training process or whether there are some starting conditions that work well in a wide range of cases. It is also possible that the Gaussian assumption on the data was misguided. While the assumption was consistent across all models, it would be interesting to look into other distributions, like for instance the multivariate student t-distribution. We would also like to experiment with the factorized version of the GWP, which we did not get around to.

We would like to also include the GMM method for a better comparison. In hindsight, it would have been better to replace the MOGP model with the graphical VAR model given the same amount of time. Both models have a static covariance, but the VAR model would have had the additional benefit of providing a baseline that is already being used in the field of psychopathology [4].

If the current project were to be expanded, we would also include more elaborate data simulations with for instance different types of dynamics for each dimension in the covariance matrix. We would also train the models on more simulations, at a minimum of 10 for each kind of simulation. Future research on dynamic network models for MDD can be expanded by including more models. One interesting candidate encountered during this research is a fairly recent (2021) support vector regression (SVR) based method for covariance forecasting [18]. This method calculates the range-based (co)variance and uses its Cholesky decomposition to train an SVR model. The SVR then forecasts the elements of those Cholesky factors which are then used to reconstruct the covariance forecast. Even without extensive optimization and exploration of the method, it surpassed DCC-MGARCH on volatility forecasting of financial time-series data using a basic linear kernel with a length scale of 15. Since the dataset used in this project also contained several measurements a day, it should be possible to apply range-based covariance estimators and the rest of the method. It would provide us with another non-parametric alternative to the GWP.

7 Acknowledgements

I want to thank my supervisor Max Hinne for guiding me throughout this process and for providing me with useful advice. Furthermore, I want to thank David Leeftink for implementing the BANNER framework in GPflow 2.0 in the first place, and Hester Huijsdens for sharing her experience with similar struggles.

References

- [1] R.H. Belmaker and Galila Agam. “Major Depressive Disorder”. In: *The New England journal of medicine* 358 (1 June 2009), pp. 55–68. ISSN: 0028-4793. DOI: [10.1056/NEJMRA073096](https://doi.org/10.1056/NEJMRA073096). URL: <https://www.nejm.org/doi/full/10.1056/nejmra073096>.
- [2] Luis Gutiérrez-Rojas et al. “Prevalence and correlates of major depressive disorder: a systematic review”. In: *Brazilian Journal of Psychiatry* 42 (6 Aug. 2020), pp. 657–672. ISSN: 1516-4446. DOI: [10.1590/1516-4446-2020-0650](https://doi.org/10.1590/1516-4446-2020-0650). URL: <http://www.scielo.br/j/rbp/a/gC5yf6KyWB7F4wBc7ChbcKv/?format=html&lang=en>.
- [3] Angélique O.J. Cramer et al. “Major Depression as a Complex Dynamic System”. In: *PloS one* 11 (12 Dec. 2016). ISSN: 1932-6203. DOI: [10.1371/JOURNAL.PONE.0167490](https://doi.org/10.1371/JOURNAL.PONE.0167490). URL: <https://pubmed.ncbi.nlm.nih.gov/27930698/>.
- [4] Sacha Epskamp et al. “The Gaussian Graphical Model in Cross-Sectional and Time-Series Data”. In: *Multivariate Behavioral Research* 53 (4 July 2018), pp. 453–480. ISSN: 00273171. DOI: [10.1080/00273171.2018.1454823](https://doi.org/10.1080/00273171.2018.1454823). URL: <https://www.tandfonline.com/doi/abs/10.1080/00273171.2018.1454823>.
- [5] Daniela Stojanova et al. “Network regression with predictive clustering trees”. In: *Data Mining and Knowledge Discovery 2012 25:2* 25 (2 June 2012), pp. 378–413. ISSN: 1573-756X. DOI: [10.1007/S10618-012-0278-6](https://doi.org/10.1007/S10618-012-0278-6). URL: <https://link.springer.com/article/10.1007/s10618-012-0278-6>.
- [6] Sadia Shakil, Chin Hui Lee, and Shella Dawn Keilholz. “Evaluation of sliding window correlation performance for characterizing dynamic functional connectivity and brain states”. In: *NeuroImage* 133 (June 2016), pp. 111–128. ISSN: 1053-8119. DOI: [10.1016/J.NEUROIMAGE.2016.02.074](https://doi.org/10.1016/J.NEUROIMAGE.2016.02.074).
- [7] Annastiina Silvennoinen and Timo Teräsvirta. “Multivariate GARCH Models”. In: *Handbook of Financial Time Series* (2009), pp. 201–229. DOI: [10.1007/978-3-540-71297-8_9](https://doi.org/10.1007/978-3-540-71297-8_9). URL: https://link.springer.com/chapter/10.1007/978-3-540-71297-8_9.
- [8] David Leeftink, Max Hinne, and Hester Huijsdens. *GitHub - DavidLeeftink/BANNER: A gpflow 2 implementation of the variational Generalized Wishart Process*. URL: <https://github.com/davidleeftink/BANNER>.
- [9] Andrew Gordon Wilson and Zoubin Ghahramani. *Generalised Wishart Processes*. URL: <http://mlg.eng.cam.ac.uk/andrew>.
- [10] *Depressive Disorders, Diagnostic and Statistical Manual of Mental Disorders*. American Psychiatric Association, 2013. DOI: [10.1176/appi.books.9780890425596.dsm04](https://doi.org/10.1176/appi.books.9780890425596.dsm04). URL: <https://psychiatryonline.org/doi/10.1176/appi.books.9780890425596.dsm04>.
- [11] Derek Richards. “Prevalence and clinical course of depression: A review”. In: *Clinical Psychology Review* 31 (7 Nov. 2011), pp. 1117–1125. ISSN: 02727358. DOI: [10.1016/j.cpr.2011.07.004](https://doi.org/10.1016/j.cpr.2011.07.004).
- [12] Nederlands Vereniging voor Psychiatrie (Dutch Psychiatry Association). *Depressie Symptomen*. Available at <https://www.nvvp.net/website/patinten-informatie/aandoeningen-/depressie/symptomen>. URL: <https://www.nvvp.net/website/patinten-informatie/aandoeningen-/depressie/symptomen>.
- [13] E. S. Paykel. “Partial remission, residual symptoms, and relapse in depression”. In: *Dialogues in Clinical Neuroscience* 10 (4 2008), pp. 431–437. ISSN: 12948322. DOI: [10.31887/dcns.2008.10.4/espaykel](https://doi.org/10.31887/dcns.2008.10.4/espaykel).
- [14] Jolanda J. Kossakowski et al. “Data from ‘Critical Slowing Down as a Personalized Early Warning Signal for Depression’”. In: *Journal of Open Psychology Data* 5 (1 Feb. 2017). ISSN: 2050-9863. DOI: [10.5334/JOPD.29](https://doi.org/10.5334/JOPD.29). URL: <http://openpsychologydata.metajnl.com/articles/10.5334/jopd.29/>.

- [15] *Venlafaxine: MedlinePlus Drug Information*. Available at <https://medlineplus.gov/druginfo/meds/a694020.html>. URL: <https://medlineplus.gov/druginfo/meds/a694020.html>.
- [16] Marieke Wichers et al. “Critical Slowing Down as a Personalized Early Warning Signal for Depression”. In: *Psychotherapy and Psychosomatics* 85 (2 Feb. 2016), pp. 114–116. ISSN: 0033-3190. DOI: [10.1159/000441458](https://doi.org/10.1159/000441458). URL: <https://www.karger.com/Article/FullText/441458%20https://www.karger.com/Article/Abstract/441458>.
- [17] Denny Borsboom. “A network theory of mental disorders”. In: *World Psychiatry* 16.1 (2017), pp. 5–13. DOI: <https://doi.org/10.1002/wps.20375>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/wps.20375>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/wps.20375>.
- [18] Piotr Fiszeder and Witold Orzeszko. “Covariance matrix forecasting using support vector regression”. In: *Applied Intelligence* 51 (10 Oct. 2021), pp. 7029–7042. ISSN: 15737497. DOI: [10.1007/s10489-021-02217-5](https://doi.org/10.1007/s10489-021-02217-5).
- [19] Creighton Heaukulani and Mark Van Der Wilk. “Scalable Bayesian dynamic covariance modeling with variational Wishart and inverse Wishart processes”. In: (2019).
- [20] Monica Billio, Massimiliano Caporin, and Michele Gobbo. “Flexible Dynamic Conditional Correlation multivariate GARCH models for asset allocation”. In: *Applied Financial Economics Letters* 2 (2 Mar. 2006), pp. 123–130. ISSN: 17446546. DOI: [10.1080/17446540500428843](https://doi.org/10.1080/17446540500428843).
- [21] Alexios Galanos. *rmgarch: Multivariate GARCH models*. R package version 1.3-6. 2019.
- [22] C. Rasmussen and C. Williams. “Gaussian Process for Machine Learning”. In: (Jan. 2006).
- [23] S Roberts et al. “Gaussian processes for time-series modelling. Gaussian processes for time-series modelling”. In: (2013). DOI: [10.1098/rsta.2011.0550](https://doi.org/10.1098/rsta.2011.0550). URL: <http://dx.doi.org/10.1098/rsta.2011.0550>.
- [24] Edwin V. Bonilla, Kian Ming A. Chai, and Christopher K. I. Williams. *Multi-task gaussian process prediction*. Dec. 2007. URL: <https://dl.acm.org/doi/10.5555/2981562.2981582>.
- [25] Alexander G. de G. Matthews et al. “GPflow: A Gaussian process library using TensorFlow”. In: *Journal of Machine Learning Research* 18.40 (Apr. 2017), pp. 1–6. URL: <http://jmlr.org/papers/v18/16-537.html>.
- [26] *A simple demonstration of coregionalization — GPflow 2.5.1 documentation*. Available at <https://gpflow.github.io/GPflow/2.5.1/notebooks/advanced/coregionalisation.html#References>.
- [27] Thang D. Bui, Cuong V. Nguyen, and Richard E. Turner. “Streaming Sparse Gaussian Process Approximations”. In: (May 2017). URL: <http://arxiv.org/abs/1705.07131>.
- [28] Eric Ghysels and Joanna Jasiak. “GARCH for Irregularly Spaced Financial Data: The ACD-GARCH Model”. In: *Studies in Nonlinear Dynamics & Econometrics* 2 (4 Jan. 1998). ISSN: 1558-3708. DOI: [10.2202/1558-3708.1035](https://doi.org/10.2202/1558-3708.1035). URL: <https://www.degruyter.com/document/doi/10.2202/1558-3708.1035/html>.
- [29] François Éric Racicot, Raymond Théoret, and Alain Coën. “Forecasting irregularly spaced UHF financial data: Realized volatility vs UHF-GARCH models”. In: *International Advances in Economic Research* 14 (1 Feb. 2008), pp. 112–124. ISSN: 10830898. DOI: [10.1007/S11294-008-9134-2](https://doi.org/10.1007/S11294-008-9134-2).