A brain-inspired cognitive support model for stress reduction based on an adaptive network model

Andrei Andrianov a, S. Sahand Mohammadi Ziabari b,⇑, Charlotte Gerritsen b

a Computational Science Lab, University of Amsterdam, Amsterdam, the Netherlands
b Social AI Group, Vrije Universiteit Amsterdam, Amsterdam, the Netherlands

Received 17 July 2020; accepted 27 October 2020
Available online 10 November 2020

Abstract

Stress is often seen as a negative factor which affects every individual’s life quality and decision making. To help avoid or deal with extreme emotions caused by an external stressor, a number of practices have been introduced. In the scope of this paper, we take three kinds of therapy into account: mindfulness, humor, and music therapy. This paper aims to see how various practices help people to cope with stress, using mathematical modelling. We present practical implementations in the form of client–server software, incorporating the computational model which describes therapy effects for overcoming stress based on quantitative neuropsychological research. The underlying network model simulates the elicitation of an extremely stressful emotion due to a strong stress-inducing event as an external stimulus, followed by a therapy practice simulation leading to a reduction of the stress level. Each simulation is based on user input and preferences, integrating a parameter tuning process; it fits a simulation for a particular user. The client–server architecture software which has been designed and developed completely fulfills this objective. It includes server part with embedded MATLAB interaction and API for client communication.

© 2020 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Keywords: Integrative model; Computational model; Temporal-causal network; Network-oriented modelling; Stress; Cognitive modelling

1. Introduction

A cognitive development of a designed model and incorporating cognitive computational models helps individuals, who experience high level of stress, to better understand their stress level and the roles of different therapies to decrease their stress level during a period of following therapy and later on a potential recommendation of changing or continuing therapy based on the performance of the selected therapy. There are some previous works on cognitive behavior of individuals with a high level of stress (Ziabari & Treur, 2018; Ziabari, 2019; de Haan et al., 2019; Ziabari, 2019; Langbroek et al., 2019; Mohammadi Ziabari & Gerritsen, 2020; Ziabari, 2019; Ziabari & Treur, 2019; Lelieveld et al., 2020).

The structure of the implemented network model including the cognitive states and connections showing the interactions between the states is justified based on the quantitative research available at the moment in the fields of Psychology, Neuroscience, Social Sciences, Computational Science. In this paper we describe these works and show the supporting evidence for the network model.
design. A large number of temporal-causal network-oriented modeling literature on therapies to decrease stress has been published recently, including papers which contain detailed descriptions of the therapies that we take into account in our research, namely: mindfulness, humor and music therapy (Andrianov et al., 2019; Ziabari & Treur, 2018, 2019, 2020). The main idea behind this type of research is to study the undergoing cognitive processes during various therapy practices on extreme emotions handling. The mechanism behind these therapies is as follows: a person, considered to be under stress due to external stimuli, starts to follow one of the therapies. After a therapy session the stress level decreases, thus helping the person to suppress extreme emotions. Another assumption which has been experimentally proved to be right is that after repeating these therapy sessions for a while, the person’s stressful feelings are not as intense as before they started practicing the therapy. Moreover, the individual does not get stressed as easily anymore, thus developing a stronger mental state. In computational modelling these effects are achieved by incorporating the effect of Hebbian learning rule (Caporale & Dan, 2008) which describes the plasticity of neural connections.

Mindfulness-based therapy practices have been successfully used for the purpose of coping with stress and various clinical applications to known extent at least since the 2000s (Davidson et al., 2003; Kabat-Zinn et al., 1998; Rückert, 2015; Särkämö et al., 2008). Afterwards, the growing body of literature showed importance of these techniques in dealing with extreme emotions by providing supporting empirical evidence (Eberth and Sedlmeier, 2012; Farb et al., 2010; Hede, 2010). These techniques include for example the so-called mindfulness-based stress reduction (MBSR) program as described by Kabat-Zinn et al. (Kabat-Zinn et al., 1998) and also various yoga and meditation techniques (Jevning et al., 1992; Scharfstein et al., 1978). The term mindfulness means essentially the self-regulation of ones’ attention, including concentration and attention switching. It is considered to be a metacognitive skill as practicing mindfulness would require both control of cognitive process (i.e., attention self-regulation) and monitoring the stream of consciousness (Bishop et al., 2004; Flavell, 1979). Although the mindfulness approach itself is often associated with practices as yoga and meditation, it may be incorporated into various other activities, for instance - autogenic training (Ziabari & Treur, 2019).

The positive effects mentioned such as stress reduction previously were studied for mindfulness and music therapy (Gotink et al., 2018; John et al., 2012; Ziabari and Treur, 2018), indicating the differences with people who do the practice and who do not. The topic has been studied from the neuroscience perspective to show the underlying neural mechanisms (Sivoňová et al., 2004). Another research in this knowledge domain aimed to study in particular brain areas related to memory, between the people who practice mindfulness regularly over a long time period and people who do not (McEwen, 1999).

The result showed significant positive correlation between mindfulness practice and the size of the memory related part of the brain, as well as higher performance in episodic and working memory related tasks among people who practice regularly (Brown et al., 2016; Hölz et al., 2011). Thus, memory interactions were also addressed and incorporated in the implemented mindfulness simulation (Andrianov, Guerriero, & Ziabari, 2019), showing the connections between different memory types and other various cognitive states during the therapy practice.

Music in general and music therapy in particular seem to be a more favored topic for research, as it is simpler to carry out different studies related to brain activity measurements (Koelsch et al., 2013). In case of mindfulness, it is indeed possible to make the measurements, but only in a pretest/post-test manner. Unlike the music therapy, where the studied individual may experience listening to music or singing while being monitored at the time. Surely, this depends on the kind of mindfulness therapy which we aim to study, however it makes it more challenging to do while a person is moving as in yoga practice, or needs to be in a certain position, including their ambience, as in meditation. Comparable to mindfulness, music therapy has also shown to positively affect memory functions, and have a great potential in neurological rehabilitation, cognitive recovery and prevention of depression as could be seen from work of Ross, Niebling, and Heckert (1999). The effects of music therapy include positive intervention for adults with symptoms of stress, depression, and anxiety (Hanser, 1985). Another research studied these positive effects of music therapy to suppress anxiety in stressful situation for patients undergoing surgical operations (Shapiro, Schwartz, & Bonner, 1998), showing that music indeed is a low-cost therapy that has a great potential in reducing not only extreme emotional feelings (Kemper & Danhauer, 2005). Moreover, music therapy firmly affects common emotion processing in such areas of the brain as Amygdala, Hippocampus and Orbitofrontal Cortex, as pointed by Scharfstein, Alon, Biderman, Daor, and Hoffmann (1978). Generally, music therapy appears as a more clear and accessible treatment for people as it has a lower entry threshold and the appreciation for music is widely shared within our society. Although the preferences in types of music varies individually and culturally, certain kinds of music occur to have more consistent effects, as showed by McCratty, Barrios-Choplin, Atkinson, and Tomasino (1998). Thus, we distinguish the persons’ reaction for “sad” and “happy” music (Biller et al., 1974; Khalifa et al., 2008) and also incorporate the effects of singing while listening to music, into the computational model.

Another therapy which is considered in the scope of this research is humor therapy. Humor is essential part of communication in our society, highly valuable for social interaction. The main effect of humor is laughter, which is considered to be the physical expression of its appreciation (Keith-Spiegel, 1972). It helps us to express our positive emotions such as feeling of happiness, excitement and
Quantitative research is available on the subject of humor and laughter, related to its effects on human both psychologically and physiologically (Berk, 2001; Biller et al., 1974; Fine, 1989; Jevning et al., 1992). The same research also shows how humor is used in communication between health care professionals and patients in hospitals to subdue anxiety, fear, anger, embarrassment and other feelings associated with stress. Compared to mindfulness, humor therapy similarly to music has a lower entry threshold for the users, and it is commonly spread in our society. Laughter corresponds with release of happiness hormones in the brain - endorphins, which showed efficiency in controlling pain response (Fine, 1989; Tse et al., 2010). Additionally, Steelman (1990) highlighted its major effects in dealing with chronic pain together with increasing happiness. Overholser (1992) pointed out in his work how humor can be used in life to deal with a variety of stressful situations. He also showed how the efficiency for such experiments might be successfully measured. Positive physiological effects of humor therapy include its impact on muscular, respiratory, circulatory, endocrine, immune, and cardiovascular systems (Berk, 2001; Fry, 1992; Tang et al., 2015). Thus, both physiological and psychological effects were incorporated in the simulation including for instance lung capacity or abdominal muscle.

In this paper, a cognitive development of a designed model is used to incorporate all above-mentioned cognitive computational models. This helps users, who experience high level of stress, to better understand their stress level and the roles of different therapies to decrease this stress during a period of following therapy and later on a potential suggestion of changing/maintain therapy based on the performance of the selected therapy in the first phase. For this, the underlying network model simulates the elicitation of an extremely stressful emotion following with a therapy simulation leading to a reduction of the stress level.

Incorporating the components which we described in this section, we can draw a detailed diagram for the ambient application which illustrates the process inside the implemented sub models in the user-ambient interface (Treur, 2016b). Fig. 1 illustrates the resulting ambient application diagram which is broken down in three main components: analysis model, support model and adaptation model.

The paper is organized as follows. In Section 2 the analysis model is described by the use of one computational model, mindfulness therapy. In Section 3 the support model of a brain-inspired cognitive architecture is presented. In Section 4 the adaptation model using parameter tuning is discussed. Ambient-Internal Stress Architecture is addressed in Section 4 of this paper and finally, Section 5 is a discussion.

2. Ambient model diagram

In this section we will discuss the ambient model diagram, as depicted in Fig. 2. The interaction between the sub models helps the user to estimate the amount of stress she/he still perceives has following therapy and a broader overview of the outcome if he/she continues the therapy with the help of simulation results in built-in cognitive computational models and using previous history of doing a therapy by the users.

Red boxes represent important features inside the application while white boxes indicate minor steps. Here the relation between all sub-components is demonstrated using arrows, the direction of arrows shows how different parts influence each other (e.g. parameter tuning with simulated annealing directly affects the customization of simulation for a user). Clear differentiation of the adaptation model is difficult as it also shares common parts with two other models. The process inside the depicted ambient application is as follows: at first a user enters his/her preferences and answers questions regarding the current situation for analysis purposes; next the therapy type is determined based on available data, which is then used for the current
user; prior to starting a simulation, a parameter tuning process is applied along with averaging to adapt the simulation; finally, simulation assessment is made after the simulation run to determine its effectiveness.

In the analysis sub model, the user inputs such as environmental stress and the internal level of stress are captured by a user input and this information is used as a basis for simulating different cognitive computational models. As has been mentioned earlier, we have considered three different cognitive computational models namely, mindfulness, music, and humor to give the user better understanding of each therapy that he/she is using.

The support sub model is used to clear up the efficiency of a therapy which has been chosen by the user, if the simulation outcome was not in a decreasing trend in the stress level considering the history and preference of a user experience in the support model, the notification of advising to change a therapy is sent to the user. This sub model will be described in Section 4.

In the following sub model, adaption model, takes place and parameter tuning using simulated annealing together with moving average method plays a role. In the adaptation sub model, based on the constraints that have been predefined in the computational model an efficient outcome is determined to be further analyzed by the support model. This will be in Section 5.

As it has been described, the aim of these models is to develop an ambient sub models with the help of cognitive computational models to support people with stress. This ensures people with high level stress to see the result of their performances on following a particular therapy and keeping or changing that therapy based on previous results. This has been done by getting a user therapy preference on doing a particular therapy and other inputs such as answering different questions regarding their beliefs on the usage of therapies to decrease the amount of stress and utilize them as the inputs of the above-mentioned sub models to give a better understanding of using those therapies and give an advice to the users to maintain or change the therapy based on the history of using computational models.

3. Analysis model

The analysis model (blue in Fig. 2), validates user input and prepares data for the domain model, and also includes an assessment of simulation results. This part of the model gathers information from the user such as age, profession, etc. The user has to fill out this information together with his/her experienced stress level and his/her preferences for using a specific type of therapy. This information is used as an input for performing a simulation at the back-end together with the output of the simulated annealing in the adaptation model and the support model. The outcome result from the analysis model is used as an input for the simulation in adaptation model. Furthermore, the user preference is updated after determination of a suggested therapy in the adaptation sub model and after that it is updated as an input to the support sub model.
3.1. Computational Model: Mindfulness as an example

Differential equation models are used in many fields such as physics, biology, economics, and engineering. A differential equation model tries to make a connection between a function to its own derivatives (ten Broeke, 2017). As stated in ten Broeke (2017) due to lack of existence of an exact solution for most differential equation models, numerical methods might be used to approximate the solution. In order to verify a temporal-causal network model one can use mathematical analysis. This will occur when a stationary point of a state occurs at some point in time. Stationary point means that there is not any change at the specific time.

Causal modeling and causal reasoning have a long tradition in science (Bentler, 1980). The Network-Oriented Modelling approach based on temporal-causal networks as described in (Treur, 2016a) could be considered a continuation of this tradition, also incorporating the dynamic and adaptive temporal perspective on mental states by means of cyclic causal connections, describing cognitive behavior patterns, as in (Kim, 2018). As it was briefly mentioned previously, the computational model implemented for this research is a temporal-causal network model which is based on the Network-Oriented Modelling approach as detailed in (Treur, 2016a). Temporal-causal network models can be portrayed by means of two interchangeable representations: conceptual representation, presented either graphically by a graph or an adjacency matrix; numerical representation, derived in the form of mathematical difference and differential equations. It is commonly used as a base for network-oriented modelling on the subject of extreme emotions (Andrianov et al., 2019; Modinos et al., 2010; Ziabari and Treur, 2020).

The network itself consists of cognitive states and connections between them that represent causal impacts of the states on each other. Each of these states has an activation level that changes value over time. As in the real world, some states are able to change their values faster than others. This is represented in the network by speed factors of the states. In the same manner the strength of a connection between states may be stronger or weaker,
which is handled by the connection weights. Furthermore, to aggregate multiple causal impacts on a state from other states the concept of a combination function is used. Thus, the three main concepts of the Network-Oriented Modelling approach based on the temporal-causal network could be described as follows:

- **Connection weight** $o_{X,Y}$ of the connection from a state $X$ to a state $Y$ represents the strength of this connection and the impact value that state $X$ has on state $Y$. This is often between 0 and 1, but sometimes below 0 for a negative effect, or above 1.

- **Combination function** $c_Y(...)$ of a state $Y$ is chosen for each state in order to aggregate the causal impacts of other states on $Y$.

- **Speed factor** $\eta_Y$ of a state $Y$ handles the speed of activation change for $Y$ upon impact from $X$.

### 3.2. Computational model overview

The proposed network model is based on the extreme emotion network adopted from previous work on the topic of mindfulness practice and coping with extreme emotions (Ziabari & Treur, 2019). Additions have been added on top of it to extend it and introduce how yoga practice affects this network, helping an individual to deal with stressful feelings.

An example of such conceptual representation, fully showing the structure of a yoga practice network model could be seen in Fig. 3. As mentioned previously the effects of mindfulness therapy on different cognitive states in this case are incorporated. The actions in yoga practice are divided into breathing and body movement, which are performed simultaneously, as derived from literature (Andrianov et al., 2019). Dotted lines represent the adaptive connections, achieved my applying the Hebbian learning rule. Table 1 illustrates summarized explanation for the considered states in the conceptual model in Fig. 1.

In Fig. 4 the simulation outcomes are presented for a person with mindfulness therapy. The upper graph corresponds with a generalized simulation, while on the bottom we see this simulation after the adaptation process for a particular use-case. X-axis indicates the time, Y-axis corresponds to the activation level for a state in the network. The test simulation was made for a user with a higher level of confidence as result of the therapy and with higher persistence in his/her goal. As we can see the adaptation process alters the results of a simulation to make it more appropriate for this user. Namely, we can notice how the feeling of extreme emotion does not increase as quickly and has more suppression in the beginning. The goal state for a therapy practice is also affected, as it seems to be less urgent in this case. The overall pattern for the extreme emotion clearly shows more smooth behavior and has a lower peak as if the person has developed some resistance for the extreme emotion.

| States | Abbreviation | Description |
|--------|--------------|-------------|
| $X_1$  | wsee         | World state for extreme emotion ee |
| $X_2$  | ssee         | Sensory state for extreme emotion ee |
| $X_3$  | wsc          | World state for context c |
| $X_4$  | sc           | Sensory state for context c |
| $X_5$  | srssee       | Sensory representation state for extreme emotion ee |
| $X_6$  | srs           | Sensory representation state for context c |
| $X_7$  | fsee         | Feeling state for extreme emotion ee |
| $X_8$  | preee        | Preparation state for extreme emotion ee |
| $X_9$  | exe          | Execution state for extreme emotion ee |
| $X_{10}$ | goal          | Goal state for yoga practice |
| $X_{11}$ | sss          | Self-awareness state |
| $X_{12}$ | sb           | Sensory state of breathing b |
| $X_{13}$ | srsb         | Sensory representation state for breathing b |
| $X_{14}$ | esb          | Execution state of breathing b |
| $X_{15}$ | memw         | Memory state for episodic memory w |
| $X_{16}$ | ins          | Body interoception state |
| $X_{17}$ | psy          | Preparation state for yoga |
| $X_{18}$ | memw         | Memory state for working memory w |
| $X_{19}$ | ssb          | Sensory state for body movement b |
| $X_{20}$ | srsb         | Sensory representation state for body movement b |
| $X_{21}$ | esb          | Execution state for body movement b |
| $X_{22}$ | memlt        | Memory state for long-term memory lt |
| $X_{23}$ | bisb         | Belief state for positive belief p |
3.3. Numerical representation: Mathematical analysis

As detailed in (Treur, 2016a, Ch.2) we transformed the proposed conceptual model representation into the numerical representation of the model for later use as basis for the simulation and further mathematical analysis. In this section the process is described and the resulting numerical representation is shown. The following applies:

- For every time point $t$ in the interval $[0, T]$ with step $\delta t$, each state $Y$ in the model has a real number value $Y(t)$ which usually lies in the interval (Andrianov et al., 2019).

- At each time point $t$ every state $X$ connected to state $Y$ has an impact on $Y$ defined as:

\[
\text{impact}_{X,Y}(t) = \omega_{X,Y} X(t),
\]

where $\omega_{X,Y}$ is the weight of the connection from $X$ to $Y$.

- The aggregated impact of multiple states $X_i$ on $Y$ at time $t$ is determined using some combination function $c_Y(\cdot)$, such that:

\[
\text{aggimpact}_Y(t) = c_Y(\text{impact}_{X_1,Y}(t), ..., \text{impact}_{X_k,Y}(t)) = c_Y(\omega_{X_1,Y} X_1(t), ..., \omega_{X_k,Y} X_k(t))
\]

where $X_i$ are the states with connections to state $Y$.

- The effect of $\text{aggimpact}_Y(t)$ on $Y$ is exerted over time gradually, depending on speed factor $\eta_Y$:

\[
Y(t + \Delta t) = Y(t) + \eta_Y \text{aggimpact}_Y(t) \Delta t
\]

or $\frac{\partial Y(t)}{\partial t} = \eta_Y \text{aggimpact}_Y(t) - Y(t)$

As mentioned previously the following combination functions $c_Y(\cdot)$ were used for the states. For states with single impact from only one other state the identity function $\text{id}(\cdot)$ was used. For states with multiple impacts from other different states the scaled sum function $\text{ssum}_\lambda(\cdot)$ with scaling factor $\lambda$ and the advanced logistic sum function $\text{alogistic}_r(\cdot)$ with steepness parameter $\tau$ and threshold parameter $\sigma$ were used as combination functions:

\[
\text{id}(V) = V
\]

\[
\text{ssum}_\lambda(V_1, ..., V_k) = (V_1 + ... + V_k)/\lambda
\]

\[
\text{alogistic}_r(V_1, ..., V_k) = \frac{r}{1 + e^{-\tau(V_1 + ... + V_k) - s}}
\]
For each state and the set of values selected, such as connection weights, speed factors and combination functions for this state, there is a difference and differential equation assigned respectively. Below follow some examples of such difference and differential equations which were used in the model. For the states with single incoming impact and identity function these formulas are shown in Table 2.

For the states which have multiple impacts from several different states with scaled sum function and advanced logistic sum function used these formulas are shown in Table 3.

As has been described earlier, this model uses a result from doing a particular therapy, and the amount of a stress on the environment and a feeling internally from the user. As has been described earlier, this model uses a result from the simulation to find an optimum value for each cognitive computational model. The output from the analysis sub model is used as an input for customized simulation.

The current support model contains three supportive actions advised by the ambient model proposed in Fig. 2 to the user. The first support action is called “checking history of previous advice” and the second one is called “check preference”. In the first step, the previous advice (if there is one) is compared to the user’s initial preference and if they are the same there would not be an advice to the user for changing the therapy.

The third support action is called “checking current situation based on available data”. In this stage, the comparison between available data, after completion of the adaptation sub model, with the user preference is happened. If the stress level over time (available simulation result/assessment) did not decrease while using a particular therapy, the suggestion is made in this sub model to inform the user to change his/her therapy.

5. Adaptation model

The adaptation model (purple color in Fig. 2) consists of simulated annealing, exponential averaging and also customization of the generalized domain model. Exponential averaging together with simulated annealing are done by getting the input such as therapy preference, beliefs on doing a particular therapy, and the amount of a stress on the environment and a feeling internally from the user. As has been described earlier, this model uses a result from the simulation to find an optimum value for each cognitive computational model. The output from the analysis sub model is used as an input for adaptation sub model in
which parameter tuning with pre-existing Monte Carlo sensitivity analysis and moving average is performed. The output of the adaptation sub model is used for the assessment of the simulation.

In this section we will discuss each sub component of the adaptation model illustrated in Fig. 2, respectively.

5.1. Parameters tuning

There are many parameter estimation methods which are methodological namely, linear squares, nonlinear least squares, sensitivity analysis, gradient decent, simulated annealing, evolutionary computing, fuzzy logic, Bayesian belief networks, and particle swarm optimization. Each of them has its own advantages and disadvantages regarding computational time, simplicity, finding a global optimum (ten Broeke, 2017).

In order to fit the outcome of model for each user the parameter tuning is used. This has been done by bound constrained minimization by the help of the simulated annealing.

5.2. Simulated annealing

In order to make the outcome of the model individual for every user, we use a parameter tuning process. Bound constrained minimization is used for this purpose by means of simulated annealing. This method is an adaptation of pre-existing Monte Carlo methods. The simulation of annealing can be used to find an approximation of a global minimum for a function with a large number of variables (Khachaturyan, Semenovsovskaya, & Vainshtein, 1981).

First, we make an initial run of the generalized model. If the parameters tuning is on for a user, we run simulated annealing for the general model, with unknown speed factors as variables for the optimization. Simulated annealing is computed with the objective function which returns the scalar value of the objective function. The objective function Root Mean of Squared Residuals (RMS) is calculated on deviations of the model values from the empirical data, in our case the result of the generalized model.

\[
RMS = \sqrt{\frac{SSR}{N}} = \sqrt{\frac{(X(t_1) - Y(t_1))^2 + \cdots + (X(t_N) - Y(t_N))^2}{N}}
\]

where \(Y(t_i)\) is the empirical data sample and \(X(t_i)\) is the value predicted by the model at time \(t\). The simulated annealing algorithm generates an arbitrary trial point and chooses the distance of the trial point from the current point by a probability distribution with a scale depending on the current temperature. We used fast annealing with step length that equals the current temperature, and direction distributed uniformly random. After generating the trial point, the algorithm moves the point in order to find the extremum of the function. As previously mentioned we set upper and lower boundaries in the range \([0.5; 0.9]\) for the algorithm to lower the number of iterations and reduce computation time. The algorithm chosen trial points are uniformly distributed at random, incorporating value at each iteration. The algorithm periodically lowers the temperature and stores the best point found to this moment. Iterations stop when the average change in the objective function is small relative to set up function tolerance. We used common tolerance value of \(1 \times 10^{-6}\) and the default acceptance function for new points. See the work of Ingber (2000) for a more detailed description of this algorithm.

\[
I = \frac{1}{1 + \exp\left(\frac{\Delta}{T_0}\right)}
\]

where \(\Delta\) is the difference between objectives, \(T_0\) initial and the current temperature. In this application we also set up another limitation. This limitation has been done by limiting the maximum computational time and setting upper boundary to 90 min. If in the worst case, computation is not finished after 90 min we get values from the best iteration. This time limitation can be withdrawn by using more powerful machines. This decision was made after an experimental run. Initially, running the simulated annealing algorithm with the indicated objective function and all speed factors of a model as variables took more than 12 h and was not done at that moment. Thus, we continued the experiments with decreasing the number of variables and concluded to use speed factors of 3 key states as variables for the optimization problem. The time boundary is set to 90 min by cause of experimental runs for 5 states which took more than 2 h to compute. Currently the 3 states considered are \(w_{se}; f_{se}\) and \(e_{se}\), which represent the world states for extreme emotion, feeling and execution states for extreme emotion respectively.

Although, the simulation tuning part was implemented using threads and asynchronous computations, it does not provide the sufficient level of parallelism needed for a more efficient computation. However, this is based on the carried experiments, which took place using a compact laptop, and the result might differ for a run on more powerful machines. Another possible improvement point, for this process is to implement optimization with various other algorithms such as gradient boosting/stochastic gradient descent to compare performance and find the best method experimentally. Gradient descent usage might be more favorable in case where high-performance computing (HPC) might be unavailable, since it could more efficiently find local optimum of a function rather than simulated annealing, which is more suited for the global optimum calculation.

5.3. Exponential moving average

As was pointed out earlier, we set up a minimum number of recent simulations in order to turn on the parameter
tuning part. Thus, to normalize the data we start optimizing the model output when at least 5 simulations took place. Prior to each simulation run, the number of recent simulations is indicated to make a decision on applying parameter tuning and averaging or not. This process is done both for the user input simulation parameters, such as stress level, positive faith, etc., and for the results of simulated annealing. For this purpose, we use exponential moving average (EMA) in order to normalize these values and achieve a smoother function behavior. Below this formula for the simulated annealing results is presented:

\[
EMA_i = \left( \eta_Y - EMA_{i-1} \right) \times \left( \frac{2}{N + 1} \right) + EMA_{i-1}
\]

where N is the number of recent simulations to take into account, and \( \eta_Y \) is the speed factor which is used as variable for simulated annealing. EMA, which is also known as an exponentially weighted moving average (EWMA) is a weighted form of moving average. It serves as an infinite impulse response (IIR) filter, applying weighting factors which decrease exponentially. Using the averaging allows us to smooth out short-term fluctuation dynamics and highlight more long-term patterns in the data, avoiding extreme situations. A number of previous values to take into account for the initial value depend on user input data. Large deviations in the input will affect the total, even for a considerably small weight. We presume the input to change gradually, and not to have great variations over time, thus we consider 5 recent simulations for averaging.

6. Ambient-internal stress interaction

In this section the architecture of the software is presented. The architecture is designed and implemented following widespread design patterns such as client–server partitioning and Model-View-Controller (MVC) designation. The client server architecture implies that the client (in our case it is an application GUI) sends requests to the server part (through API made with Java). The server processes request and sends response back to the client, which then communicates it to the user. MVC patterns help in separating internal information representations from the way it interacts with the user. The view component is presented as a front-end GUI. While the back-end has models, which serve as dynamic data structures, and controllers, which map requests, validate the input data and perform interactions on the data model objects. In Fig. 5 the interaction between these components is depicted.

We annotated models as Java Persistence API (JPA) entities. JPA is used as a layer that handles the management of relational data in the application by communicating with the database.
6.1. Overview of an interaction

During the application design process, functional and architecturally-significant requirements were first developed, together with the use-case diagram as shown below in Fig. 6. In the use-case diagram the functionality of the software is depicted from the user perspective as available actions. Each action is shown in a separate ellipse. The order of the actions is depicted using arrows. The loops inside the use-case diagram emphasize continuous actions which may be performed at any time since first usage.

Together with the depicted ambient architecture it gives a complete overview of the implemented software. As for the technological stack for the development, it was concluded to use the Java programming language for the back-end server part mainly because it is a current industry standard and there is extensive supporting material available. We used Spring Boot framework for building the server and restful API due to the same reasons, also it runs on Java EE and provides great collections for building web applications. We used MATLAB for the entire computational part of the project as it naturally handles mathematical implementations, computational models and provides powerful optimized vector operations. In addition, the recently published MATLAB engine API for Java provides a powerful communication layer which simplifies interaction between the two and perfectly suits our implementation case. As mentioned previously we used client–server...
software architecture with the client being a Java Script GUI application on a mobile device and server incorporating restful API, implemented using Java and Spring Boot framework.

Figs. 7–9 show the framework of the cognitive-built software. It starts with signing/logging into an account using a g-mail account and it follows with providing the age, profession, name, and the gender of the user. In the profession section we have considered seven categories that almost cover most of the professions in which the user can choose. In the next slide of this framework the user needs to provide some information about the experienced stress which is used as an input for the back-end of the software. The progress slide shows the stress level of the user as a his-
tory. And last but least, with the provided disclaimer we informed users that this software is not a treatment but a more IT-based stress-related software to show the progress based on different therapies.

6.2. Server and API

The server part is divided as mentioned, following MVC principles. It includes two models being User and Simulation, and corresponding User and Simulation controllers. Thus, these models are used as data entities and the controllers are implementing all the data manipulations through corresponding repositories, as well as the mapping of the API calls. Another class implemented next to the models and controllers is the simulation service class which contains all the logic regarding various simulation manipulations, such as suggestion of therapy type or running the simulation. This service when called opens connection to a MATLAB Engine Future instance using the MATLAB Engine API for Java, then it communicates with the engine by sending data, running commands and receiving the results. All the interaction between Java and MATLAB is implemented using asynchronous calls, thus making it possible to run multiple tasks simultaneously. Below in Fig. 10 the component diagram is shown, depicting the software architecture as distinct components.

Results of every simulation run and parameter tuning process are saved as a MATLAB native binary file, which contains the data in an array form. Moreover, the user input parameters (i.e. initial values for a simulation) are saved in a comma-separated values (csv) format for each simulation run together with the results of simulated annealing process. This allows us to easily check the parameters when there is no need in operating MATLAB.

The API on the server has in total nine different method functions implemented, including http GET, PUT and POST methods. The front-end part makes calls to these API methods, server handles the logic and returns result back to the front-end. These methods include data handling for basic user information, their preferences regard-
which has been designed and developed completely fulfills adaptation parts. The client–server architecture software integrative model incorporating analysis, support and system, e.g. correct email format for user id. Another implemented check covers the input data validation and communication the check results with the client side. In both cases custom exceptions were thrown if the user with a specified id exists in the database. As a single simulation is connected to only one user. All the implemented API methods were tested thoroughly and a number or validation techniques were put inside the controller classes. Including checks for existing users while saving new user data, to avoid duplicating entries in the database, as well as in case of a GET call to the API we check if the user with a specified id exists in the database. In both cases custom exceptions were made to communicate the check results with the client side. Another implemented check covers the input data validation, e.g. correct email format for user id.

7. Discussion

The initial objective of the project was to develop an integrative model incorporating analysis, support and adaptation parts. The client–server architecture software which has been designed and developed completely fulfills this objective. It includes a server part with embedded MATLAB interaction and API for client communication. The final version of the software meets all architecturally significant requirements which were set in the beginning. Prior to the simulation and software development, numerous supportive literature and previous research has been studied extensively, main points of interest in the design process and the development have been described in previous chapters. The goal of the final application, to serve as an interface for user interaction with the model, along with gathering empirical data to support further research has been achieved. This made users aware of their performances during using a particular therapy when they were under stress. These results from interaction between different sub models namely, the analysis model, support model, and adaptation model, help the people with a high level of stress to keep or change their therapy based on their historical performances and built-in cognitive computational models.

Discussing the limitations of this work, first of all we would like to point out the computational time for the simulation and parameter tuning part - as it was mentioned previously in order to achieve the greater efficiency for the algorithms, it’s worth considering usage of other HPC techniques apart from threading and asynchronous calls used in the scope of this research. Currently, parallel execution is only feasible for about ~15–20% of the code, increasing this number may lead to a significant improvement in performance. However, this needs to be done together with hardware-level parallelism such as distributed computing, to achieve the best outcome. Thus, the results of the optimized simulation output heavily depend on the amount of variables and parameters for the simulation process which in turn define individualistic outcomes of a simulation. Results of this research open up numerous possibilities for further studies as the implemented piece of software may be further used to study the effects of various therapies on stressed individuals.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Andrianov, A., Guerriero, E., & Ziabari, S. S. M. (2019). Cognitive modeling of mindfulness therapy: effect of yoga on overcoming stress. In International symposium on distributed computing and artificial intelligence (pp. 79–86). Cham: Springer.

Bentler, P. M. (1980). Multivariate analysis with latent variables: Causal modeling. Annual Review of Psychology, 31(1), 419–456.

Berk, R. A. (2001). The active ingredients in humor: Psychophysiological benefits and risks for older adults. Educational Gerontology, 27(3-4), 323–339.

Biller, J. D., Olson, P. J., & Breen, T. (1974). The effect of “happy” versus “sad” music and participation on anxiety. Journal of Music Therapy.
Ziabari, S. S. M., & Treur, J. (2018). Computational analysis of gender differences in coping with extreme stressful emotions. *Procedia Computer Science, 145*, 376–385.

Ziabari, S. S. M., & Treur, J. (2018). An adaptive cognitive temporal-causal network model of a mindfulness therapy based on music. In *International conference on intelligent human computer interaction* (pp. 180–193). Cham: Springer.

Ziabari, S. S. M., & Treur, J. (2019). Cognitive modeling of mindfulness therapy by autogenic training. In *Information systems design and intelligent applications* (pp. 53–66). Singapore: Springer.

Ziabari, S. S. M., & Treur, J. (2019). An adaptive cognitive temporal-causal model for extreme emotion extinction using psilocybin. In *Proceedings of the computational methods in systems and software* (pp. 176–186). Cham: Springer.

Ziabari, S. S. M., & Treur, J. (2020). An adaptive cognitive temporal-causal network model of a mindfulness therapy based on humor. In *Information systems and neuroscience* (pp. 189–201). Cham: Springer.

Ziabari, S. S. M., & Treur, J. (2020). Decision making under acute stress modeled by an adaptive temporal-causal network model. *Vietnam Journal of Computer Science*. 