

Deliverable 4.4

“An open source-based simulation available on the web that demonstrates the strengths of the models as well as simulation results”

Lead Partner:	University of Skovde (HIS)
Authors:	Jonas Mellin
Date:	March 2015
Revision:	V1.0
Dissemination Level	PUBLIC



With the support of



Project Acronym:

HELICOPTER

Project full title:

HEalthy Life support through COmPrehensive Tracking of individual and Environmental Behaviors

AAL project number:

AAL-2012-5-150

TABLE OF CONTENT

1. Verification and Validation of anomaly detection	3
1.1 Feature experiment of anomaly detection	3
<hr/>	
2. Factors	3
3. General simulation model for the experiment	4
3.1 Circadian rhythm.....	5
3.2 Sleep state model.....	6
3.3 The disruptor model.....	10
<hr/>	
4. Results	11
4.1 Disruption	12
<hr/>	
5. Discussion	16
5.1 Parameter setting/configuration.....	16
5.2 Data representation	17
5.3 Validity of results	17
<hr/>	
6. Conclusions	17
6.1 Future work.....	17
<hr/>	
7. References	18
8. Validity threats	18



With the support of



1. Verification and Validation of anomaly detection

A vital part of any development is verify and validate the artefacts that has been constructed. In this report, verification implies that we build the product right and validation implies that we build the right product. The emphasis is on the former, but the latter is addressed to since we, for example, need to reduce false positives to 1 in 100 evaluations of diagnostic suspicions delivered indirectly or directly to the providee.

The anomaly detection is employed to discover evidence of changed behavior over time; if evidence for changed behavior is discovered, then this evidence is reported to the hypothesis testing that takes this and other evidence into account. If the hypothesis testing concludes that there is a risk, then the caregiver is notified so that they can take action and the providee can be notified or view this if they wish to. Since the anomaly detection is incremental and local, it may adapt itself to abnormal situations according to experts but the anomaly detector accepts the abnormal behavior as normal. Therefore, the boundary value checking is required and is based on expert knowledge alone; the boundary value checking does not adapt and should contain situations that according to experts are abnormal (e.g., visiting the toilet 6-7 times or more per day).

1.1 Feature experiment of anomaly detection

The purpose of this experiment is to check that expected features of the anomaly detection are found. The expected features are:

1. If the monitored process has a majority of regular processes, then the normal model will adapt over time and in the beginning there will be a lot of false positive anomalies. After a while, this dies down and a certain amount of noise will be seen.
2. When the monitored process is disrupted, this disruption will be visible in the anomaly detection.

2. Factors

There are a number of factors that has to be addressed in simulations such as:

- Internal body clock 23-25 hours
- Sleep stage behavior.
- A virtual subject have some control over their sleeping behavior.



In this experiment, only the body clock is changed, not various thresholds for, for example, switching between different stages.

3. General simulation model for the experiment

The simulation model consists of three parts conceptually: population generation, daily life simulation and anomaly detection processing. In the population generation, different virtual subjects are configured by employing a Monte Carlo simulation. In the current model, the only thing that is set is when a disruption should occur. In the daily life simulation, two months of daily life with the focus on sleeping patterns are simulated. Finally, in the anomaly detection processing, various forms of configurations are tested. The reason for this separation is to simplify validation by reducing variance. Further, different parts of the simulation models employ separate random number generators (Law & Kelton 1991), which further reduce the variance and allows to compare features of different simulations.

The general simulation model of the daily life is based on time-stepped simulation since it is simpler to validate the model compared to event-triggered simulation. The reason is that we can validate the rules as if time stopped for a while, that is, we only need to consider the state at a given time and check if the rules represent the real world scenarios in a sufficiently useful and meaningful way.

The general representation adopted in this simulation engine is fuzzy rules (event/guard/action) organized as parallel state automatas that are grouped into a hierarchy of state machine groups, where each rule is represented by a set of transitions from state to state. At each time step, the current outgoing transitions' guards that have been triggered by the event are checked and probabilities are computed based on what transitions are applicable (i.e., their guard returns true) and the current state of the state machines in the simulation model. These transitions are performed atomically. Ordering of transitions is based on a data flow specification transformed into priorities while executing. State machines on the same priority level can be evaluated in parallel.

The generic rule set is as follows:

1. Compute bodily functions such as energy expenditure, thirsts, tiredness etc. by using as good formulaes or approximations thereof that are available in scientific literature.
2. Advance time

The anomaly detection does the following every third simulated hour:

1. Get outliers.
2. For each outlier, do
 - a. Check if it is above the upper median of its k nearest neighbours. If it is, then increase number of anomalies pointing to an increasing behavior.
 - b. Check if it is below the lower median of its k nearest neighbors. If it is, then increase number of anomalies point to an decreasing behavior.



- c. Keep anomalies in the last 24 hours classified according to in which hour relative to the current hour it is discovered.
- d. If number of increased anomalies are larger than number of decreasing anomaly, then signal evidence for an anomaly, otherwise do not.

The API for the simulation engine is found at <https://github.com/JonasMikaelMellin/AALEnvSimulator>, where the simulation models addressed in this report are found at <https://github.com/JonasMikaelMellin/AALEnvSimulator/blob/master/AALEnvSim/HELICOPTERSim/test/dailyRhythmTest/DailyRhythm3.java> (without disruption), <https://github.com/JonasMikaelMellin/AALEnvSimulator/blob/master/AALEnvSim/HELICOPTERSim/test/dailyRhythmTest/DailyRhythm3b.java> (with disruption) and <https://github.com/JonasMikaelMellin/AALEnvSimulator/blob/master/AALEnvSim/HELICOPTERSim/test/replayDailyrhythm3v2/ReplayDailyRhythm3.java> (for performing the anomaly detection). All are centered around a database implemented in MySQL where <https://github.com/JonasMikaelMellin/AALEnvSimulator/blob/master/AALEnvSim/HELICOPTERSim/test/createTables.sql> includes code for creation of the database structure.

3.1 Circadian rhythm

The circadian rhythm model (in Figure 1) is an attempt to represent the idea that the virtual subject have concerning the time of day and what is typical or acceptable to do during these periods. In particular, evening_night, night and night_morning are considered to be sleep time where probabilities are adjusted to remain in sleep and not wake up as easily as during non-sleep time. Essentially, all stay in state transitions are always true, whereas the transfer transition to the next state is based on the virtual subject’s length of a day (varying between 23-25 hours). Essentially, the virtual subject’s day length is divided by number of states to obtain the minimum length of being in a state. This is used in the guard in the transfer transitions. When a transfer transition is applicable, then the probability for transferring to the next state increases linearly for each time step to account for some variations in the virtual subject’s perception of the day.



With the support of



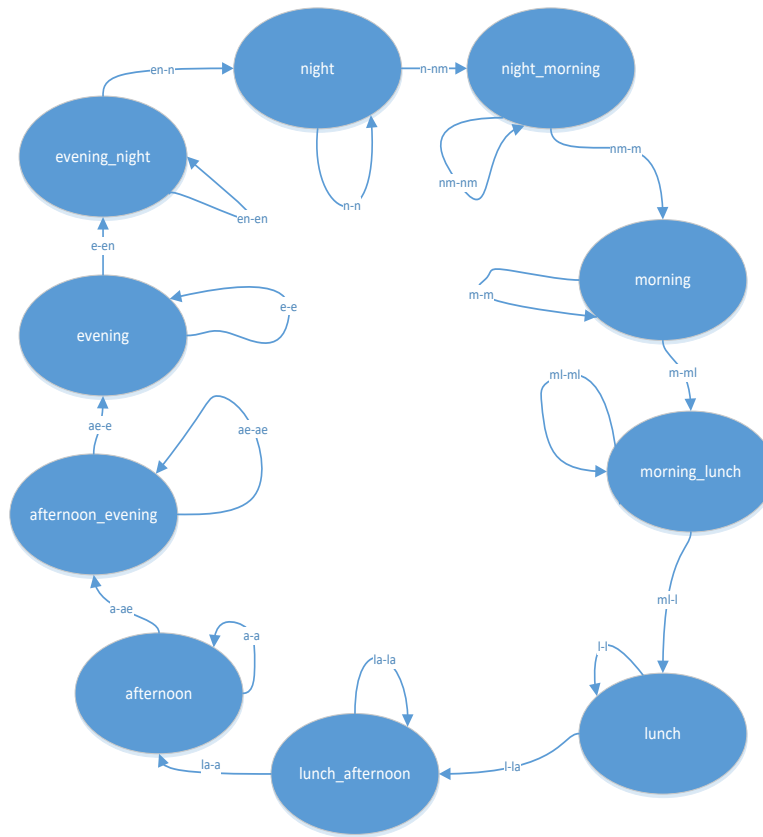


Figure 1 Circadian rhythm

The gradual increase in probability is implemented as a counted in class DayCycleProbabilityFunction starting on line 311 in <https://github.com/JonasMikaelMellin/AALEnvSimulator/blob/0.1/AALEnvSim/HELICOPTERSimulator/dailyRhythmTest/DailyRhythm3b.java>.

3.2 Sleep state model

The sleep state model (in Figure 2) captures known research on sleeping (NIGMS 2012; Wikipedia n.d.; Anon 2007). While the virtual subject is awake, it stays awake. When the virtual subject reach a sufficient fatigue level, then the likelihood for going to sleep increases. When the virtual subject is asleep, the model cycles through stage 1 to REM repeatedly while decreasing the fatigue level. As the fatigue level reaches sufficiently low levels and the body clock is in not sleep time, the likelihood for waking up increases. This waking up likelihood are adjusted for the sleep state, where it is much harder to wake up during sleep state 3 and 4 compared to sleep state 1, 2 and REM. While the person is awake, the fatigue level increases until the person is sufficiently fatigued to fall asleep. In



With the support of



this simulation model, there is no active conscious decision making, it is merely “simulated instincts”.

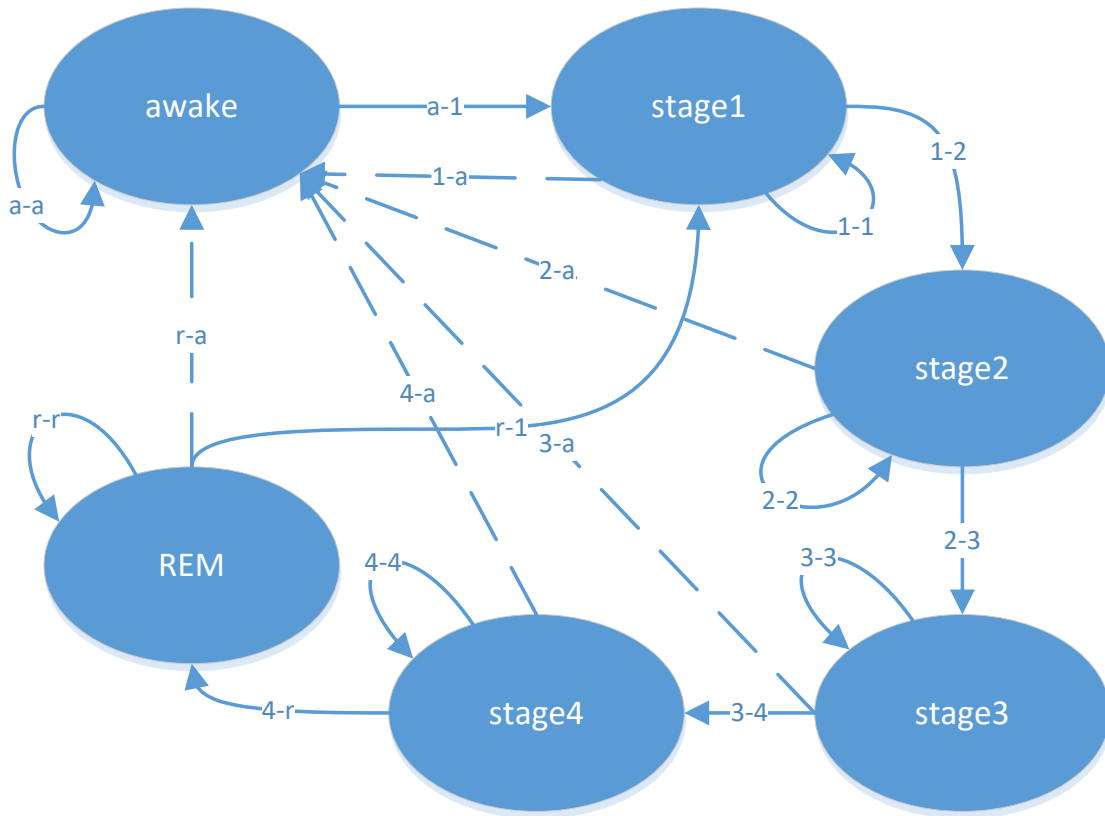


Figure 2 Sleep stages

The rules for the transitions are found in Table 1, where there is always a transition back to itself (e.g., “a-a”, “1-1”) and one going to the next sleep stage (e.g., “1-2”, “2-3”). Further, there are transitions from all stages to awake (e.g., “1-a”, “2-a”). The fatigue adjustment is positive when the virtual subject is awake and negative when the subject is asleep. These adjustments are based on the body clock, the circadian rhythm allowing for different individual length of days, different chronotypes etc. Note that these probabilities have not been fully validated. Only face and content validity have been ensured (cf. (Pitchforth & Mengersen 2013)).



With the support of



Id	Start	End	Event	Guard	Action
a-a	Awake	Awake	Tick	True	Adjust fatigue
a-1	Awake	Stage1	Tick	True	Adjust fatigue
1-1	Stage1	Stage1	Tick	True	Adjust fatigue
1-2	Stage1	Stage2	Tick	True	Adjust fatigue
2-2	Stage2	Stage2	Tick	True	Adjust fatigue
2-3	Stage2	Stage3	Tick	True	Adjust fatigue
3-3	Stage3	Stage3	Tick	True	Adjust fatigue
3-4	Stage3	Stage4	Tick	True	Adjust fatigue
4-4	Stage4	Stage4	Tick	True	Adjust fatigue
4-r	Stage4	REM	Tick	True	Adjust fatigue
r-r	REM	REM	Tick	True	Adjust fatigue
r-1	REM	Stage1	Tick	True	Adjust fatigue
1-a	Stage1	Awake	Tick	True	Adjust fatigue
2-a	Stage2	Awake	Tick	True	Adjust fatigue
3-a	Stage3	Awake	Tick	True	Adjust fatigue



With the support of



Id	Start	End	Event	Guard	Action
4-a	Stage4	Awake	Tick	True	Adjust fatigue

Table 1 Sleep state transition rules



With the support of



The adjustments of the fatigue are computed as follows:

$$night = 8 + (awakeDuration - 17) * 0,5$$

Id	Adjustment
Awake	$\frac{1 - \epsilon}{awakeDuration * 60}$
Stage1	$-\frac{1}{night * 60}$
Stage2	$-\frac{1}{night * 60}$
Stage3	$-\frac{1}{night * 60}$
Stage4	$-\frac{1}{night * 60}$
REM	$-\frac{1}{night * 60}$

At line 359 in <https://github.com/JonasMikaelMellin/AALEnvSimulator/blob/0.1/AALEnvSim/HELICOPTERSim/test/dailyRhythmTest/DailyRhythm3b.java>, the class SleepStateProbabilityFunction encapsulates the computation of the probabilities. Note that this function need to be refined and validated, for now it demonstrates irregular patterns of sleep that are probably not valid for a majority of any population. The virtual subject does not lose the circadian rhythm, but the variations are large. The major design decision in the probability computation is to provide probabilities that shows the typical behavior of staying in a sleep state for a time and then move to the next, where the variations are small.

3.3 The disruptor model

The disruptor is a simple model that awaits a particular time when the disruption is meant to happen. When the disruption occurs, configuration parameters are adjusted to represent the disruption. In this case, the length of the awoken period is increased and the sleep time is decreased.



4. Results

As can be seen in Figure 3, the average anomaly indicator based on 6 trials are, as expected, larger in the beginning before the process achieved a steady state as well as that the normal model of anomaly detection has been sufficiently adapted to the process. The anomaly detection indication decreases that remains steady with a variance as can be seen in Figure 4.

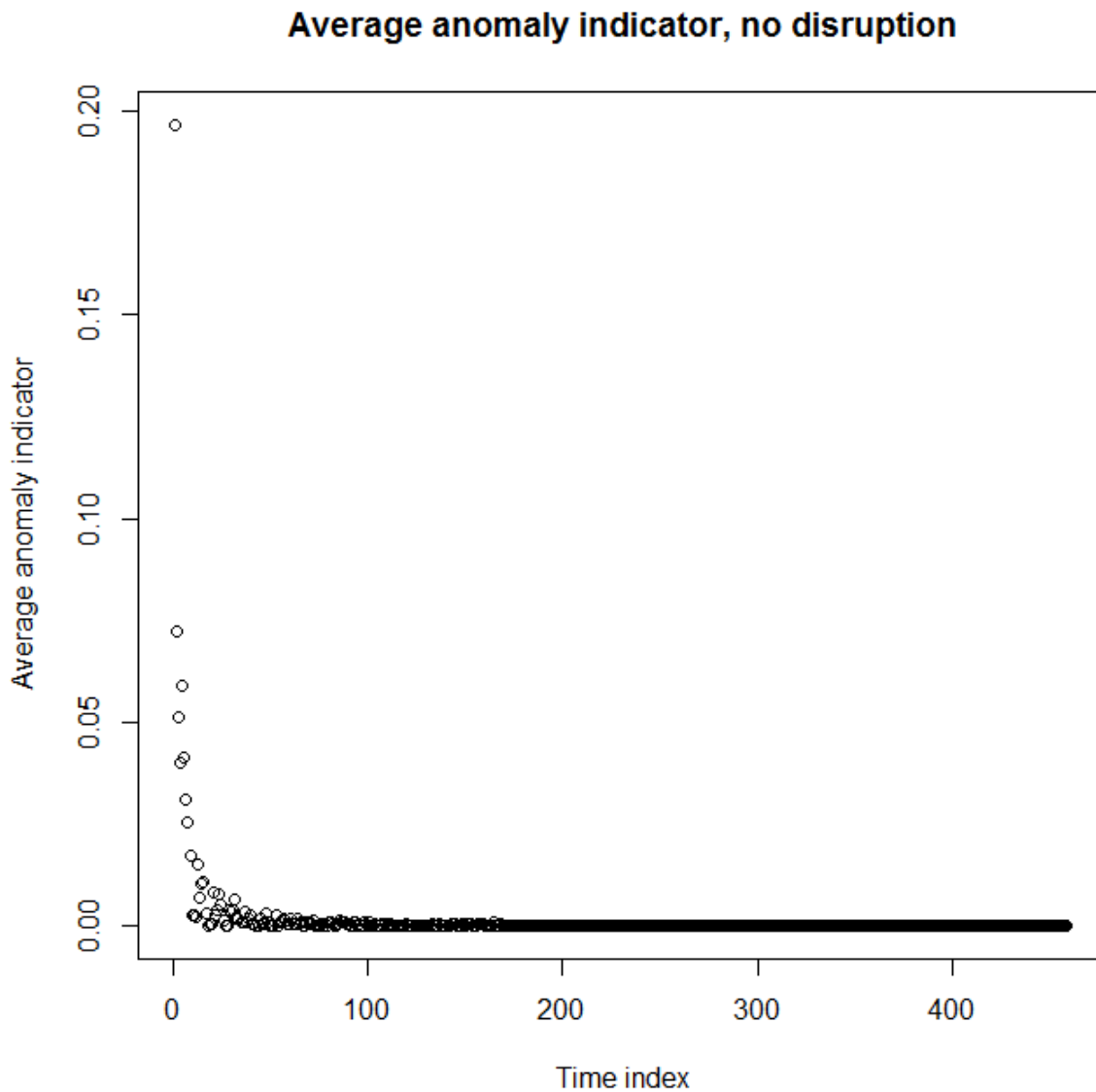


Figure 3 Average anomaly detection index of 6 trials over time, no disruption



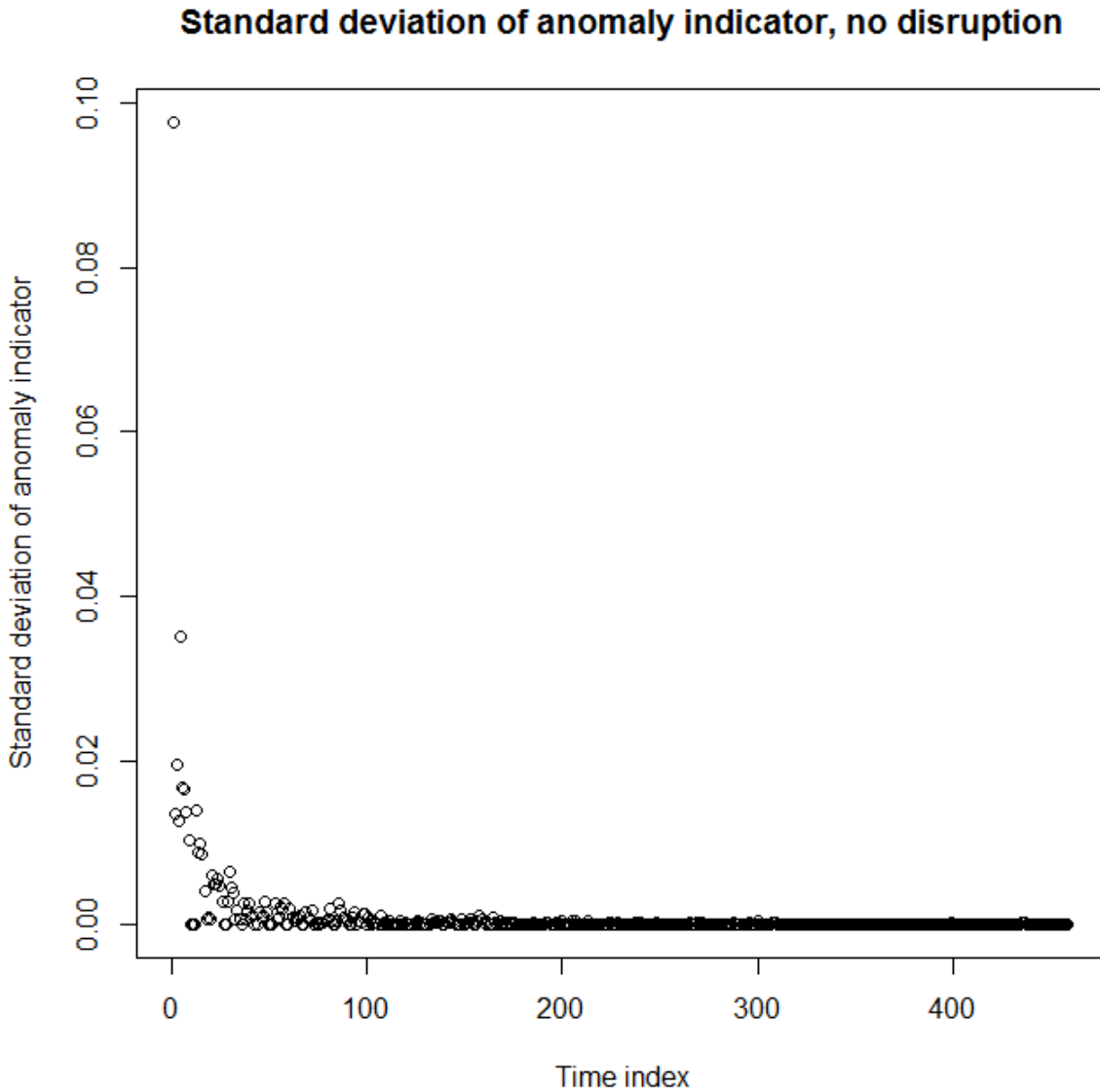


Figure 4 Standard deviation of anomaly indicator index

4.1 Disruption

At a disruption, we see more anomalies in the temporal vicinity of the disruption. Compare and contrast Figure 4 and Figure 6, where the former contains a disruption that occurs at time index 154 and the latter is the same virtual subject without a disruption in their behavior. As can be seen, there is an increased amount of anomalies between time index 154 to 200. Note that it is hard to set a threshold that on the one hand observes the anomaly and avoids false positives. The potential causes are addressed in the discussion. The few other trials show the same features, see Figure 7



With the support of



and Figure 8 for seed 1. The horizontal gray lines are the average and average plus standard deviation for non-disrupted simulations when the simulation has settled in a steady state with a normal model in the anomaly detection.

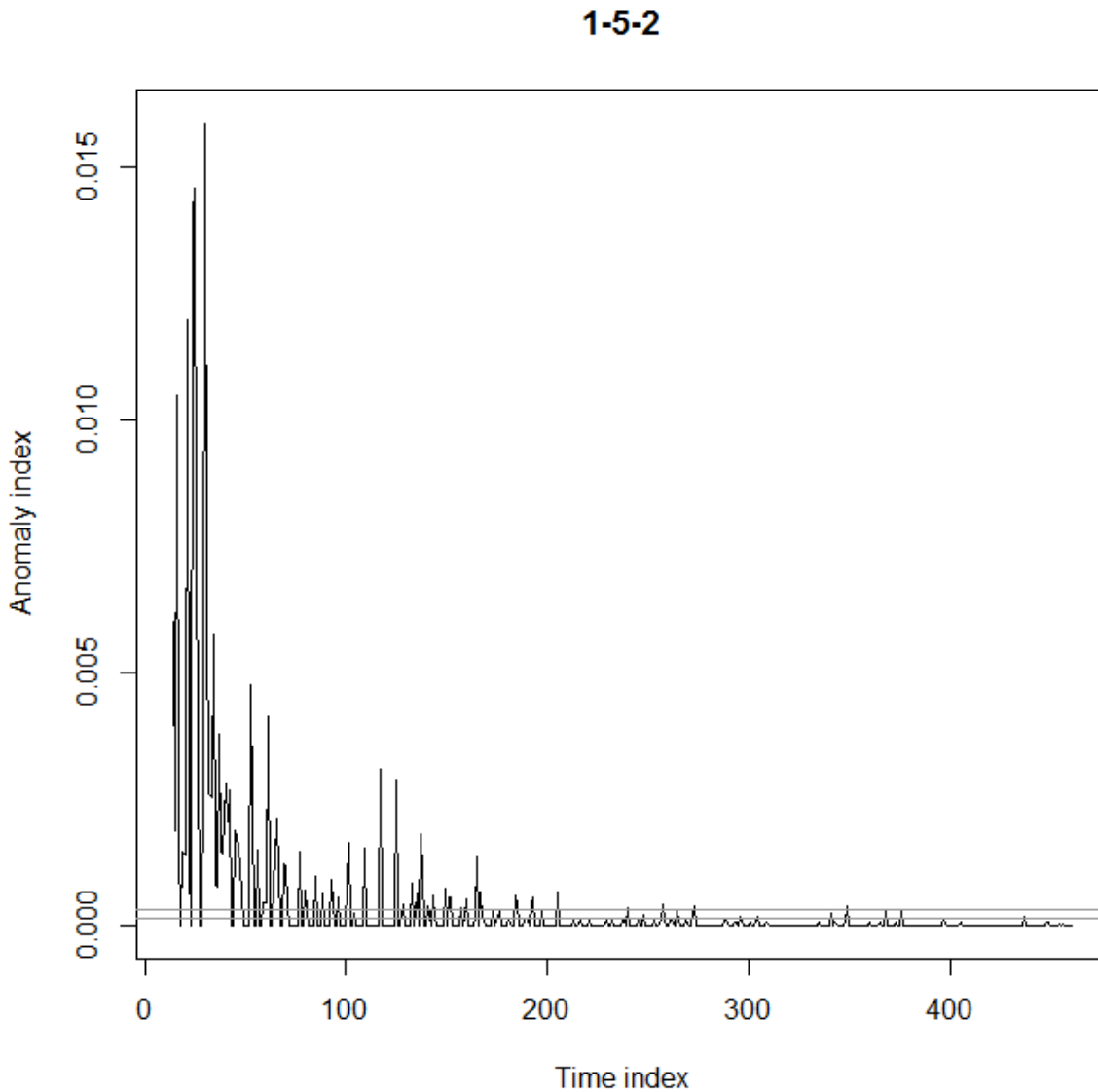


Figure 5 Trial with seed 5, disruption at time index 154



With the support of



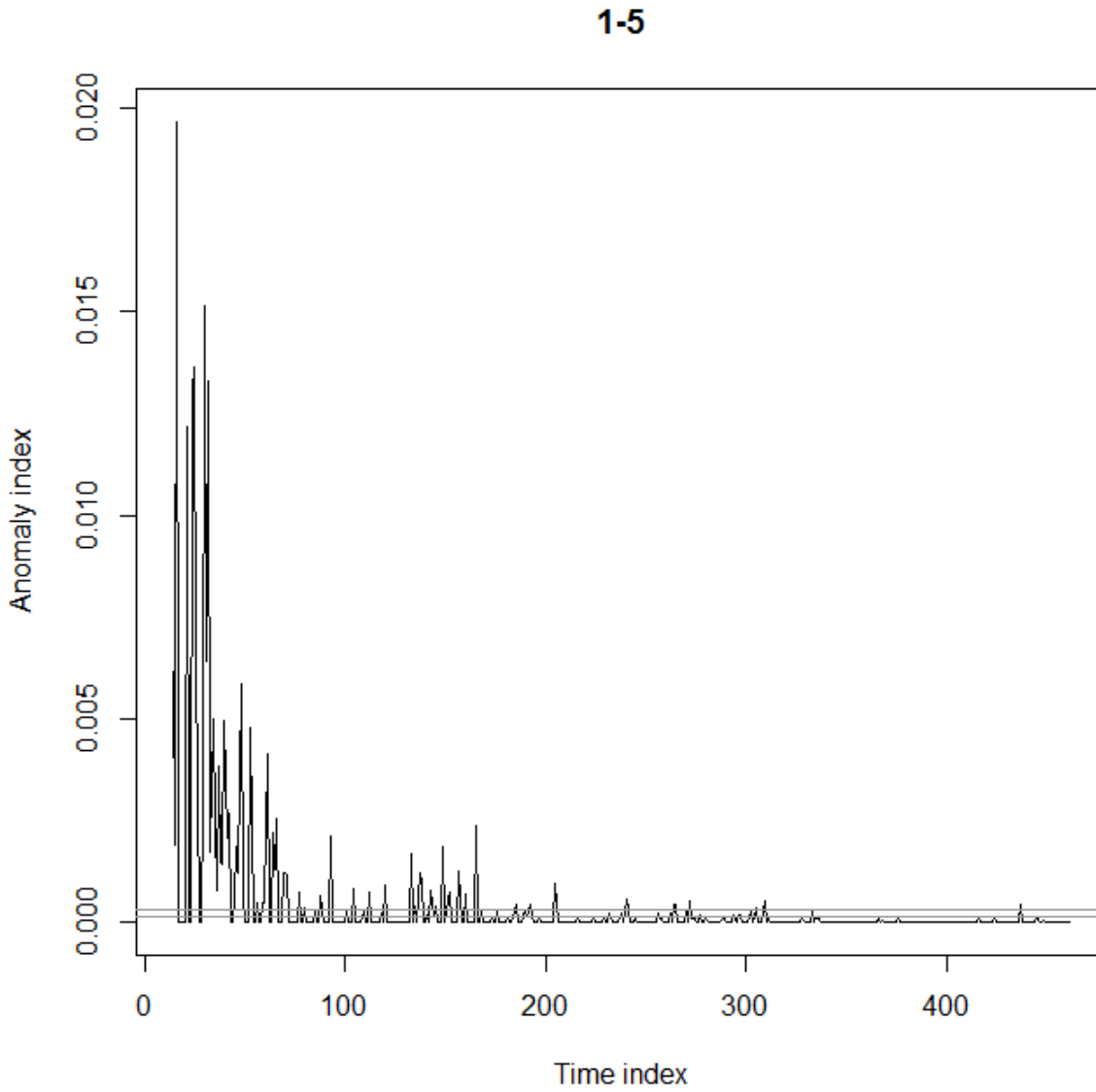


Figure 6 Trial with seed 5, no disruption



With the support of



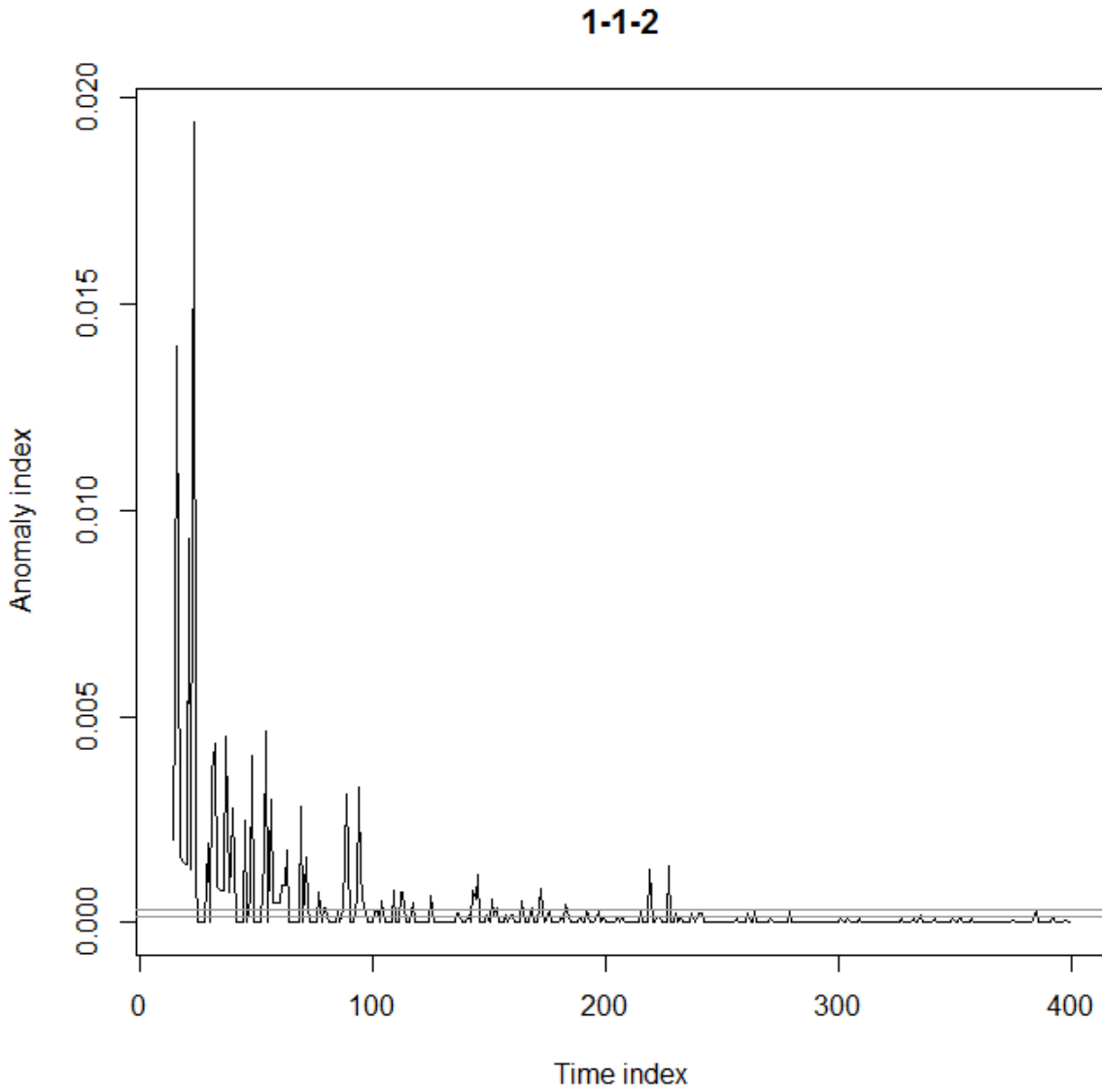


Figure 7 Trial with seed 1, with disruption at time index 116



With the support of



1-1

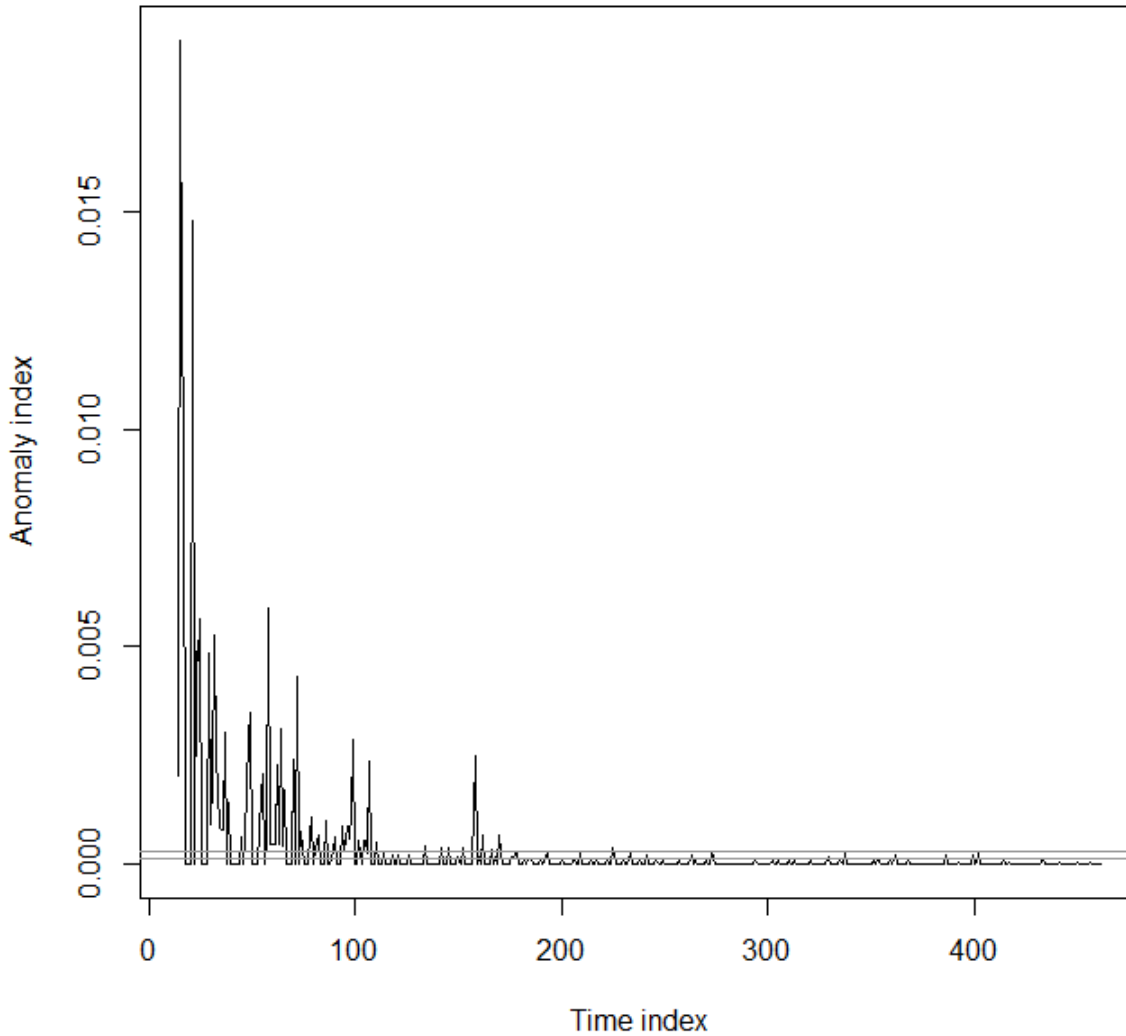


Figure 8 Trial with seed 1, no disruption

5. Discussion

5.1 Parameter setting/configuration

Essentially, the only advice provided for configuration of the anomaly detection is that $k=10$ (Pokrajac, Lazarevic & Latecki 2007). However, little is known on how to, for example, determine if there is an anomaly. Determining this threshold is an optimization problem.



With the support of



5.2 Data representation

Currently, all data sent into the anomaly detection is normalized into the range 0 to 1 in a linear transformation and distances are computed in a Euclidian space. For example, it may be better to employ a different distance computation for timestamps such as $\sum_{i=1}^n \begin{cases} d_i^2 & d_i \text{ is not a timestamp} \\ (\ln(1 - d_i))^2 & d_i \text{ is a timestamp} \end{cases}$, which would accentuate values that are close each other and penalize values that are more distant.

5.3 Validity of results

Unfortunately, the results have low statistical power due to too few trials executed and the model of sleeping behavior does not generate a particularly regular sleeping behavior. Even though this is true, it points out a number of problem as well as address methodological issues for resolving these problems.

6. Conclusions

It is concluded that the simulation model represents some kind of normal behavior since the anomaly detection stabilizes and detect fewer anomalies as time passes. Further, it is concluded from feature experiments that disruption causes anomalies compared to running the same person without the disruption. Finally, it is concluded that determining the threshold for discovering anomalies is an optimization problem, in which various configurations of anomaly detection is applied to simulated daily lifes with and without disruptions.

6.1 Future work

To determine thresholds for anomaly detection, the simulation model must be made more configurable to allow more variation in the population of virtual subjects. Further, the simulation model must be validated in its predictive capacity of simulating real sleeping behavior. The current behavior is far too varied for a normal sleeping behavior. Investigating various parameter settings to check which reproduce as normal behavior as possible so that model configuration such as the one employed in this report can be employed in a more controlled way.



Since the experiments takes so long time, it is strongly desirable to design a database as well as build a service for automating the simulation.

7. References

Anon 2007, ‘Natural Patterns of Sleep’, Harvard Medical School, viewed <<http://healthysleep.med.harvard.edu/healthy/science/what/sleep-patterns-rem-nrem>>.

Law, AM & Kelton, WD 1991, *Simulation modeling and analysis*, McGraw-Hill, New york.

NIGMS 2012, ‘Circadian Rhythms Fact Sheet’, viewed <https://www.nigms.nih.gov/Education/Pages/Factsheet_CircadianRhythms.aspx>.

Pitchforth, J & Mengersen, K 2013, ‘A proposed validation framework for expert elicited Bayesian Networks’, *Expert Systems with Applications*, vol. 40, no. 1, pp. 162–167.

Pokrajac, D, Lazarevic, A & Latecki, LJ 2007, ‘Incremental Local Outlier Detection for Data Streams’, *Proceeding of Computational Intelligence and Data Mining, 2007. CIDM 2007.*, pp. 504–515.

Wikipedia n.d., ‘Circadian rhythm’, viewed 30 November, 2016, <https://en.wikipedia.org/wiki/Circadian_rhythm>.

8. Validity threats

Validity threat category	Validity threat	Applicable?	Prevent/handle?	Notes
Conclusion	Low statistical power	Yes	Prevent	Ensure that sufficient number of trials are performed and verify that it is possible to employ statistical method prior to analysis.
	Violated assumptions of statistical tests	Yes	Prevent	Ensure that all requirements for applicable tests that we want to use are fulfilled.
	Fishing and error rate	Yes	Prevent	Be as transparent as possible and describe aim, motivation, objectives, hypothesis etc. in an as explicit way as possible without leaving out results that may be viewed as contradicting the usefulness of the approach.
	Reliability of measures	Yes	Prevent/handle	In simulation, this is prevented. In collection of real measurements, they



With the support of



Validity threat category	Validity threat	Applicable?	Prevent/handle?	Notes
				are analyzed with respect to how reasonable the measurements are.
	Reliability of treatment implementation.	Yes	Prevent/handle	In case of simulations, the simulation model must be validated. In case of real experiments, this is checked by experimental leaders.
	Random irrelevancies in experimental setting	Yes	Prevent/handle	In case of simulation, this can be prevented by appropriate model validation. In case of real experiments, this is accounted for by trying to remove them and then perform post-processing of collected data (e.g., removing data that obviously does not belong to normal behavior such as long intervals of absence when the participants should be part of the experiment). If data is modified, this should of course be reported.
	Random heterogeneity of subjects	Yes	Handle	The selection is limited in various ways and very hard to change. We can handle this in the analysis of the data.
Internal validity	History	Yes	Prevent/Handle	In case of simulation, there is no wear-out due to history. In case of experiments, subjects must be handled with care.
	Maturation	Yes	Prevent/Handle	See history. Effects of subjects that participate in multiple experiments must be addressed in the analysis.
	Testing	No		We are testing the technology itself, in some cases the experiments are with developers of the technology rather than with the providees.
	Instrumentation	Yes	Prevent/handle	We are addressing the sensor technique itself, so the instrumentation and its validation is a natural part of the development and use of the sensors themselves. In case of simulation, this is part of simulation model.
	Statistical regression	No		Too few subjects.
	Selection	Yes	Handle	Got to live with this and use available subjects when necessary.



With the support of



Validity threat category	Validity threat	Applicable?	Prevent/handle?	Notes
	Mortality	Unlikely	Handle	In simulation, not a problem. In real experiments, can be if it is quantitative analysis.
	Ambiguity about direction of causal influence	Can be	Prevent/handle	In simulation, if this is ambiguous, then either make it ambiguous or choose. In real experiments, such situations exists (e.g., does hyperglycemia cause a person to drink soft drinks or does the drinking of soft drinks cause hyperglycemia for diabetic people?). If it is not necessary to test this, then the general rule is to avoid it. Otherwise, trials catering for both directions are necessary.
	Multiple group threats	Can be	Prevent	The groups of subjects are only loosely connected and therefore the risk is not a major factor if there is a need to run experiments with multiple groups.
Construct validity	Inadequate preoperational explication of constructs	Yes	Prevent	Employ peer reviewing from different domains to reduce this risk.
	Mono-operation bias	Yes	Prevent/handle	In simulation, the idea is to employ meta-modelling to be able to generate populations and specify general rules for how the simulated population behaves in a Monte Carlo simulation according to the meta-attributes. Outliers will then be analyzed to see if the setting is realistic or if it is due to that the individual in the simulated population is unrealistic and the result can be discarded. In case of real subjects, the experiments must be much more limited and we have to be careful about interpreting the results.
	Mono-method bias	Yes	Prevention	Multiple methods employed in HELICOPTER: user experience design, experiments, case studies, simulation.
	Confounding constructs and level of constructs	Yes	Prevention	It is sensor technology that is tested in these experiments. Therefore, we do not really question real subjects except about simple things such as were you absent or present at the time.



With the support of



Validity threat category	Validity threat	Applicable?	Prevent/handle?	Notes
	Interaction of different treatments	Yes	Prevention	In real experiments, test different sensor technology separately. Further, as a general rule, let the treatments be part of daily tasks.
	Restricted generalizability across constructs	Yes	Prevention	Test things separately first, then together.
	Hypothesis guessing.	Yes	Prevent	In case of subjects that are developers, tell them the hypothesis and then ask them act as normal.
	Evaluation apprehension	Yes	Prevent	Subjects are mainly developers who like to test new technology.
	Experimenter expectancies	Yes	Handle	Peer-reviewing.
External validity		Yes		We are not testing a theory, rather we are exploring relationships.



With the support of

