

NEURAL NETWORKS FOR SMART HOMES AND ENERGY EFFICIENCY

ROESSLER, F.; TEICH, T. & FRANKE, S.

Abstract: *In a project to reduce energy consumption, the use of technology which helps metering and controlling lifestyle effects is essential. Smart meters and intelligent systems that contribute to environmental awareness enable private homeowners or tenants to see and actively control their cost of lifestyle. As a part of smart home systems neural networks are considered to be of assistance for user-based systems and consumption prediction. The observation of collected data over a period of time offers many opportunities to discover potential applications that help optimizing specific tasks. Controlling the heating process at a specific time of day, based on the habits and preferences of a tenant is one first chosen way to make daily life easier and at the same time to make it possible to design smart homes that compromise between energy-efficiency and personal comfort. For that purpose a neural network is designed and tested under varying premises. The results are promising and the insights will enable future works in following projects.*

Key words: *neural networks, learning, energy-efficiency, smart homes*



Authors' data: Roessler, F[alko]; Prof. Dr. rer. pol. habil. Teich, T[obias]; Franke, S[usan]; Westsächsische Hochschule Zwickau (UAS), Dr.-Friedrichs-Ring 2A, 08056, Zwickau, Germany, falko.roessler@fh-zwickau.de, tobias.teich@fh-zwickau.de, susan.franke@fh-zwickau.de

This Publication has to be referred as: Roessler, F[alko]; Teich, T[obias] & Franke, S[usan] (2012). Neural Networks for Smart Homes and Energy Efficiency, Chapter 26 in DAAAM International Scientific Book 2012, pp. 305-314, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-901509-86-5, ISSN 1726-9687, Vienna, Austria

DOI: 10.2507/daaam.scibook.2012.26

1. Introduction

This work contributes to a research project at the UAS Zwickau in collaboration with a communal housing association. Various sensors in newly refurbished tenements capture data of energy consumption at radiators and bus bars as well as air quality, temperature and weather conditions (Teich et al., 2010). At the present time tenants have the possibility to adjust the desired target temperature for every single room via a central touch panel that can also be timed beforehand. The heating system is configured for the number of tenements in the respective part of the building and independent from the tenants' systems. It solely serves the purpose of regulating the heat flow within the heating system as necessary and is not influenced by the effects of this work. The objective for the current research and the author's work is how to train and implement neural networks that can assist users in creating room-based heating profiles. To decrease energy loss in the form of waste heat, these profiles play a major role. As one natural adaptation the data from sensors and the smart home system can be used to train a neural network on the tenant's preferences, which then will automatically adjust the starting time for heating. As part of a smart home system, neural networks enable highly personalized services on a basis of accessible data for profiling and context sensitive actions.

2. Intention: Energy savings in private homes

Because of increasing costs for electrical energy and fossil fuels, private home owners and tenants have to address possible ways to decrease energy consumption or increase the efficiency. While electrical energy consumption depends both on the amount of electronic devices and their respective consumption, energy used for heating mainly depends on heat insulation and otherwise is constant. For that reason it does make sense to increase energy-efficiency by controlling the heating process and temperature limitations as both an upper and a lower boundary. In terms of in-room climate, conditions change with the weather and the season. These conditions influence the duration of heating a room until the preferred target temperature is reached. A set time of the start leads to either loss of energy because the room has to be kept at a certain temperature level for too long or to a time lag between the time the temperature is wanted and the time, when it is actually reached. Conditions that influence the heating duration are, for example, the temperature outside and the difference between present and target temperature as well as weather conditions. Data are collected from sensors in the building and the outside environment. The reference smart homes have been equipped with a KNX bus system on which basis most of today's standard sensors communicate. Via the central touch panel, functions of connected devices can be controlled and observed. Sensor data are sent to a local facility server in adjustable cycles. Programmed algorithms retrieve these data to a second central server over night. From that point on, procedures are necessary to filter and pre-process the data basis (Teich et al., 2010).

Besides the primary personalization there are additional problems this work is dealing with. These are in particular the automation of procedures which create, train

and evaluate neural networks as well as providing all necessary interfaces between the control system and the operating system in which the trained neural network is embedded. The objective includes a system that needs a minimal amount of manual maintenance once it is running and will re-train its neural network component, if necessary, without human intervention.

2. 1 Limitations of this work

Designing the automated processes and implementing neural networks-based programs as services into the architecture is the major task in order to provide their promising functionality in a larger scale. At the present time, the conceptual work of the automated processes is finished, but more specific personnel resources are necessary to go further into the programming and implementation process. The work and research of neural networks for the time being is limited to the specific scenario of room temperature control, with future perspectives to more small and easy-to-use applications within smart home projects. The (sub-) systems will play a supporting role for energy saving concepts until further opportunities unveil. The author's research, until this point, does not include work other than the neural network related design and the conceptual work on the automated processes, but it will blend into the software architecture design that follows in later sections and is done by another part of the related research group of the UAS Zwickau.

3. Neural networks for learning systems

Biologically inspired procedures such as genetic algorithms, ant colony optimization and neural networks (Kramer, 2009) apply to areas where traditional mathematical methods cannot be realized due to the fact that there are not enough resources to find an exact solution. These procedures mimic the way of cooperation and information processing that exists in nature. Having (artificial) neural networks, there is one abstract method comparable to the processing capabilities of the human brain. In modern practice they are used as parts of subsystems in algorithmic frameworks because they are not suitable to provide an overall system (Deco & Schuermann, 2001). Yet they can perform complex tasks ranging from recognition to short-term projection.

Neural networks consist of a minimum of two layers of neurons which each has an input, a body and an output part. In case there are only two layers, the first, also called input layer will get the input directly from external data, so there is no input connection from other neurons. A neuron will generate an output level from its total activation, i.e. the sum of net input, which is then forwarded to the connected neurons in the next layer. The output level may be kept the same value as input for the following neurons, though a multiplication with a variable factor is usually the case. Neurons in the last layer are called output neurons. Their activation levels represent the result of the 'thinking' process. They usually do not have an output connection to another unit or layer. There can be several layers between the first and last one, so called hidden layers. Neurons of the hidden layers are also called hidden units which can have many different functions and characteristics, for instance they can be used

as a form of short-term memory buffer which enables the network to make better predictions, in case there are reappearing patterns.

In detail there are many different types of neural networks (Haykin, 1994) which are not necessarily restricted to one specific function. Teaching a net is possible in different ways (Zell, 2000) which are in general divided in methods of supervised and unsupervised learning.

The field of theoretical and practical applications of artificial neural networks is vast. The work by (Leake et al., 2006) applies case-based-reasoning using neural networks for smart homes, especially eldercare, health-care and emergency solutions. An approach to energy-efficiency in wireless sensor networks is made in (Enami et al., 2010), which is limited to energy consumption of a sensor nodes network itself. Other works address forecasting models and decision systems for more specific use, e.g. electrical load forecasting or energy efficiency in production plants. While a great amount of research and practical applications have much in common on the theoretical level and the approach, there are currently no concepts of how to support energy savings in the private living sector for small but highly personalized tasks such as heating. Especially in Germany there are no public projects concerning energy-efficiency in private homes using neural networks known in the studies of the VDE, i.e. Association for Electrical, Electronic & Information Technologies (Bothoff et al., 2011).

3.1 Learning the scenario

The learning objective for the neural network is to “estimate” the time necessary to reach the target temperature which is programmed into the room control system. Data of previous heating duration are used as a basis for estimation. Since the time needed to raise temperature to a certain level differs as conditions change, these dimensions have to be included in the neural network design. These circumstances are weather conditions like wind speed and outside temperature. Internal conditions of the building complex can be taken into consideration as well, like heating system performance or status, e.g. night mode. The neural network then learns past results of heating duration in regard to temperature level difference under the respective conditions. The output of the network is forwarded to an overriding program which will also read the room’s timer and then subtracts the output. The result of that operation concludes the starting time for heating.

4. Course of action

Although the first approach to building a stable neural network has to be done manually, the objective of this work is the automation of most of the included tasks in order to create solutions for individual requirements. The following steps represent the logical order for working manually, but they do not necessarily represent the order within automated processes as they can partially be done simultaneously by the same or different computer systems and servers.

Before creating a neural network and processing data, the scenario of the required application has to be well considered. That includes: (1) the kind of output,

i.e. logical or numeric; (2) interpretation of the output by a (sub-) system or user; (3) required and available data; (4) the necessity of creating or pre-processing (artificial) data; (5) the spread of possible outputs and (6) assuring a stable solution.

4.1 Manual approach

The second step after creating the scenario is to collect all the necessary available data from the central server. After reorganizing the data, they are pre-processed into a format that the neural network can compute. It is absolutely necessary to ensure that not only the input data used for training the network but also all possible data that may occur in test an real data are within the limit of [-1 to 1] or [0 to 1] respectively.

After all data dimensions are known, the neural network architecture is built. For beginners or manual work, there are freeware tools for creating neural networks. The experiments for this research were made with the freeware tool MemBrain, a graphical neural networks editor and simulator by Thomas Jetter (Rey & Wender, 2011; www.membrain-nn.de) which allows designing simple networks. The networks that have proven suitable for the scenario at hand consists mainly of input units in the amount of dimensions in the input vector and one output unit as well as several hidden units in multiple layers.

Step four implies the creation of a set of training data as well as one or multiple data sets for testing the quality of the training process. Creating the training data set requires a representative variety of realistic data. Without that, the neural network may not be able to build up abstraction levels and to reproduce them correctly under real conditions. For that reason, it is also necessary to examine the mapping by representing the test data set(s) to the trained neural network. Neural networks possess the feature that, in some cases, they are able to generate reasonable outputs with deviant or even missing data, thus, show some sort of stability.

If, even after handling the data selection, training and test processes with high diligence, the neural network is not able to reproduce reasonable output or does not show the desired behavior, an iteration of the scenario mapping and the following steps is necessary. In some cases it may be necessary to reconsider the whole scenario, whereas adding new (artificial) data dimensions or altering existing data (e.g. in the scale) can also show the desired results. In special cases, it is also thinkable to redesign the topology of the very network in order to include logical differentiation.

4.2 First results

The first models used a six dimensional input vector containing data of temperatures, time and weather conditions over a time frame of two months within the heating period. A supervised neural network with multiple hidden layers already showed good results after a short training period.

Individual tests were iterated to redesign the network and data in order to find out which setup shows the best characteristics. If different tests had similar results the networks were analyzed in terms of performance and which would be easier to implement into a system. Training a net was done with data containing two years

measured four times per day. For the time being it is planned to design neural networks to learn on short-term basis and extend training data as they arise. First tests showed that there is a need to add more dimensions to the input vector in order to have reliable output prediction under influence of seasonal specifics. It could be achieved by adding new (artificial) parameters that were expected to harmonize transitions between seasons. A subsequent decision was to disregard non-heating seasons since they affect the learning success for the whole set of data. Experiments with additional data showed different results, yet a final commitment has still to be made.

For this kind of scenario more than only one possible solution was found as a design for a neural network. A simple recurrent network (SRN) proved to be just as effective as feed-forward networks, yet the choice of an appropriate activation function depends on the number of dimensions as well as output data requirements. While the training results for two-month data were quite similar for the output neuron activation function of the types 'tangens hyperbolicus' and 'identical' the net showed better performance in terms of speed and accuracy with the latter activation function when learning higher dimensional input vectors and a higher quantity of data. When adding new rooms to the system it will multiply the total data volume by the count which then will also prolong the required training time. Re-training a net will be done only if there are greater divergences over a certain period.

4.3. Designing the automated processes

The main tasks in order to enable automated algorithms for programs that will allow easy-to-use applications and customization for many different tenants are as follows. An overriding program (i) is needed to execute all the necessary functions, including sending commands to the neural network component core (ii), which in turn incorporates methods of creating individual neural networks. These networks are then the basis of the active element (iii) within a running program in the smart home system (fig. 1).

The overriding program prepares the data needed for the learning process. It executes the queries for both internal and external data sources via configured interfaces. Criteria for selecting as well as filtering the data are made beforehand and will be implemented as a fixed part within the algorithm. After a request, the data are consolidated and complete. From these data, training and test sets are created. Before further steps with training and test data, a network has to be created by the neural network component. That includes the definition of the necessary neuron types as well as their numbers, activation functions and connections to other units. It is also necessary to choose the teaching algorithm and to configure the condition when the teacher will stop and save the current network. As soon as the network is created and training data are prepared, the actual training process starts. Progress is evaluated by the "net error" and reference results with one or multiple sets of test data. Output tables for the test data sets are created with the help of an interface that connects the neural network component and the overriding program. If the actual output data meet the defined criteria for "success", the network is saved and will be exported to a subprogram which contains the active element and integrates it into room control

functions. In the other case, the override program will reset the network and start the training algorithm anew. Failing that process multiple times in a row, the program sends notifications to a person, who is in charge of the neural network service and then analyzes the cause. Under special conditions it may be necessary to re-evaluate the type of network and the data basis or format. The network is then redesigned and the training process starts again.

Once a neural network becomes part of the subsystem for room control and transfers to the operating mode, the active element retrieves the timer for the tenant's heating preferences. At that point, the neural network has learned to start the heating process at the right time in order to assure that the target temperature is reached at the set time, thus minimizing heat loss that occurs by maintaining the temperature level over a longer period. The active element supervises the live feedback by the tenant as well as the "live success", i.e. the divergences of the time of the desired target temperature and when it is actually reached. If there is a significant deviation in time, the neural network is transferred back to the training mode, where it is trained with updated training and test data, following the entire process from that point anew. The live feedback allows the tenant to change time or temperature manually, which is then regarded by the active element for the next actions. Re-training the neural networks will take place in the nighttime, to avoid conflicts with the active room based temperature-control (sub-) program.

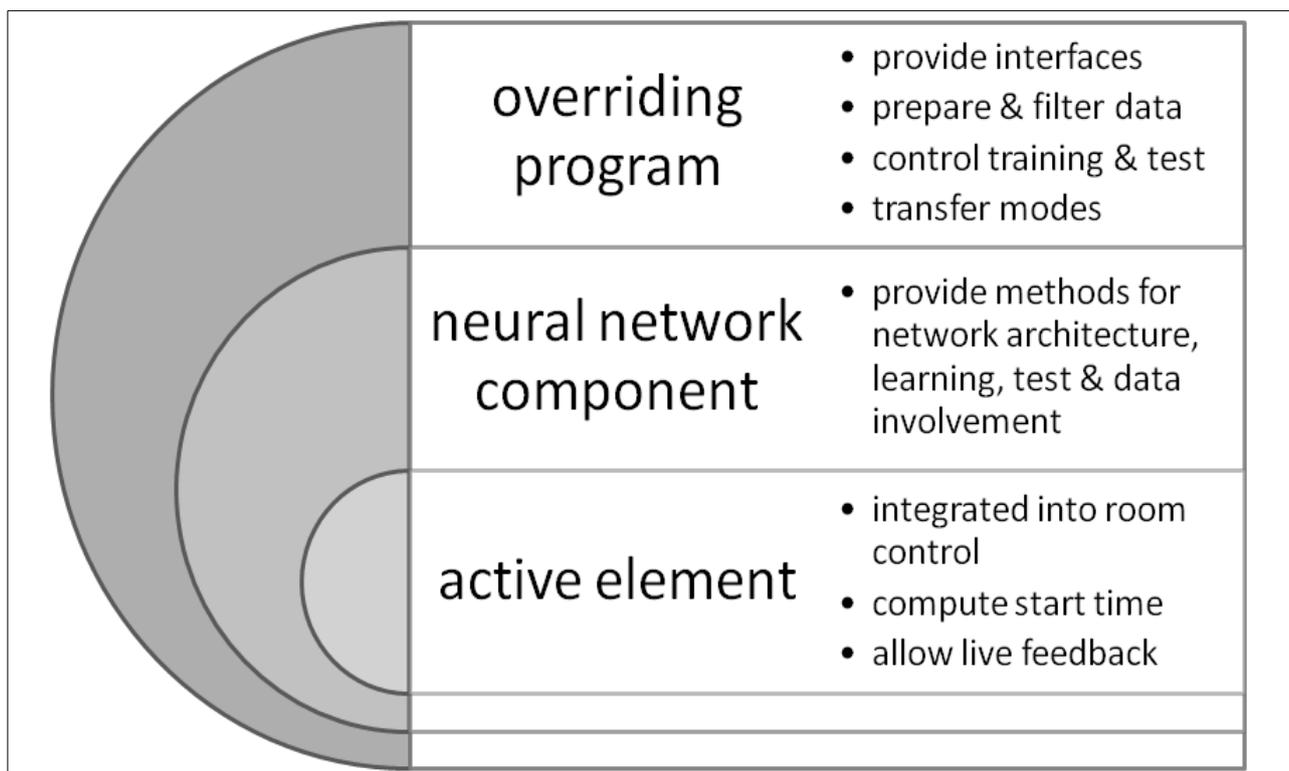


Fig. 1. Structure of the automated process design

4.4 Integration into a smart home environment for energy efficiency

Installed hardware devices like sensors, actuators and interactive control panels in the refurbished tenements provide a fundamental basis for realizing energy efficient automation solutions. Higher energy savings can be achieved by taking

entire building complexes instead of single private home areas into consideration. Traditionally, that requires information exchange among the different devices, which also means, that all of these have to use the same language. If not, cost intensive steps like additional converters and configuration efforts are necessary for interconnecting the devices. Although using manufacturer-specific properties may lighten the workload at the beginning, it is considered very risky, because it causes non-reusable software components in consequence. Eventually, the installation of new devices from different vendors with similar functionality or integrating new bus systems results in an unpredictable refactoring effort of existing software. In order to react to these kinds of changes, it is imperative to design software services and program modules as flexible as possible, to be able to adapt them as needed. The work of another project within the research group at the UAS Zwickau deals with that very problem.

The flexibility of the architecture results from the application of an OSGi platform, which is a dynamic Java based application platform. The general architecture of the software framework is shown in (fig. 2). That framework supports modularization of the software by a component model and a service registry as well as by restrictive implementation requirements. It allows collecting services and service implementations into bundles as well as configuration in a modular way (McAffer et al., 2010). Developing application logic that can be used in different system environments with distinct automation technology, it is also necessary to decouple service implementation and underlying hardware devices as well as bus systems. This results in the devices as an abstract unit and the mapping into a uniform virtual representation. Abstract devices do not contain any fixed functionality but behave like data structures. Decisions or interaction logic is implemented solely by services and business processes. A hardware-software-bridge contains adaptors responsible for the transformation of vendor specific hardware components in the technological infrastructure into the device virtualization layer and vice versa. Device specific attributes and parameters are mapped into generalized data structures of virtual proxy objects. This is done in a transformation process by hardware-specific adaptors. By extracting any device specific application logic into the service layer, time-consuming hardware programming, which is redundant for similar types of devices, can be avoided. The same method can then be used to install (sub-) programs that use the neural network as a core component for regulation room temperature. Tenants and homeowners will be able to use this service by demand, which allows them to automate their preferred comfort room-bases temperature profiles.

Generalizing hardware components from the technological infrastructure into standardized virtual objects makes it possible to develop completely reusable services as well as business processes. Services can then provide the same functionality for arbitrary devices of an identical category that was traditionally redundant and programmed into hardware devices individually. Neural networks will be a part of programs and services within that architecture. The kind of overriding program described in the chapter above is one possibility for providing individual services in a smart home environment. The system for room control receives information about the starting time of the heating process from the subprogram that runs the neural

network. Energy efficiency of the heating process on the level of the entire building complex has been optimized by proper insulation and by performing the dynamic hydraulic balance in the building (Teich et al., 2011).

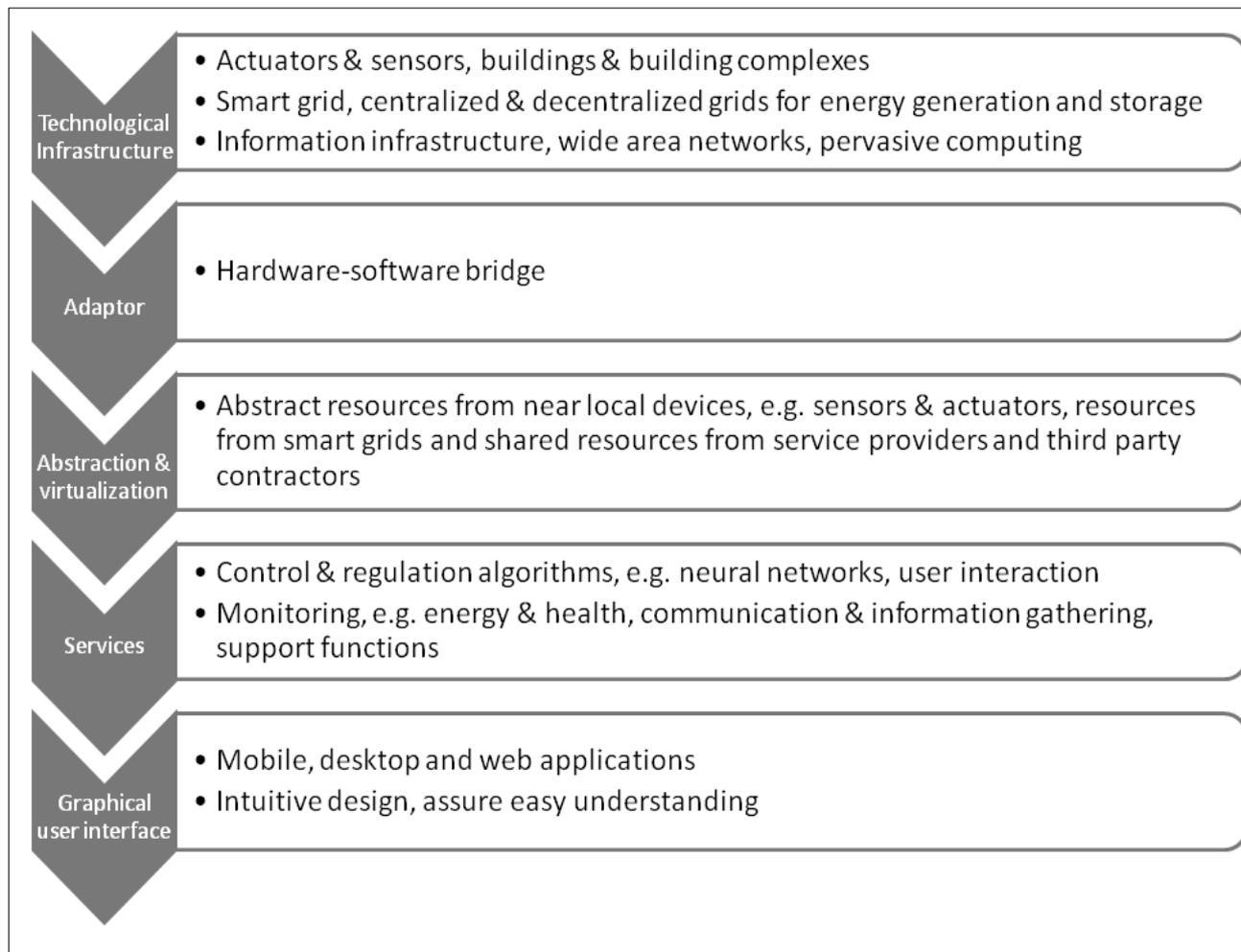


Fig. 2. General architecture of the virtualization platform

5. Results and future work

Neural networks enable the construction of various predictive systems as well as analyzing the effect of different data in specific situations. The results of the manual approach to build neural networks showed the potency to build stable solutions. The neural network produced reasonable output values that correspond to recorded data of real conditions. It is considered that the timed heating scenario is the first one which will be implemented within the next heating period and further research is going to be conducted in order to find other useful applications for neural networks in smart homes. For the time being it is a very promising approach for different situations where learning is possible or even required in the first place.

Future work includes the actual programming and implementation process according to the automation design. The author's work will be to support that process, concerning data and user interfaces as well as neural network design. Further testing and modification of the neural networks architecture will go along with

implementing the service for different users. The integration process into the service architecture will represent a major milestone to developing new smart home subsystems. The ability to provide personalized functions by simply learning from user specific preferences and behavior is a huge factor that simplifies both the daily life of tenants or homeowners and the work implied by providing that service. Possible extensions for room control services include air quality, e.g. CO₂ concentration levels or air moisture to provide more personal comfort.

6. References

- Bothhof, A.; Domroese, W. & Gross, W. (2011). *Technologische und wirtschaftliche Perspektiven Deutschlands durch die Konvergenz der elektronischen Medien*, technical report. VDI/VDE Innovation + Technik GmbH, Berlin, Germany
- Deco, G. & Schuermann, B. (2001), *Information Dynamics: Foundations and Applications*, Springer, ISBN 0-387-95047-8, Berlin Heidelberg, Germany
- Enami, N.; Moghadam, R. A.; Dadashtabar, K. & Hoseini, M. (2010). Neural Network Based Energy Efficiency in Wireless Sensor Networks: A Survey. *International Journal of Computer Science & Engineering Survey*, Vol. 1, No. 1, August 2010, pp. 39-55, DOI : 10.5121/ijcses.2010.1104
- Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation 1st*, Prentice Hall PTR, ISBN 0-023-52761-7, Upper Saddle River, NJ, USA
- Kramer, O. (2009). *Computational Intelligence*, Springer-Verlag, K., ISBN 978-3-540-79738-8, Berlin Heidelberg, Germany
- Leake, D.; Maguitman, A. & Reichherzer, T. (2006). Cases, Context, and Comfort: Opportunities for Case-Based Reasoning in Smart Homes. *Lecture Notes in Computer Science*, Vol. 4008 July 2006, pp. 109-131, ISSN 0302-9743
- McAffer, J., Vanderlei, P. & Archer, S. (2010). *OSGi and Equinox: Creating Highly Modular Java Systems*, Addison-Wesley, ISBN: 978-0321585714
- Rey, G. & Wender, K. (2011). *Neuronale Netze: Eine Einführung in die Grundlagen, Anwendungen und Datenauswertung*, Verlag Hans Huber, ISBN 987-3-456-84881-5, Bern, Germany
- Teich, T., Zimmermann, M. and Other (2010). *Intelligent Building Automation*. International Conference on Automation, Robotics and Control Systems, Karras, D.A., Moustafa K.A.F., Tang, D. (Eds.), pp. 53-57. ISRST. Orlando, Florida
- Teich, T., Szendrei, D., Schrader, M., Jahn, F. & Franke, S. (2011). *Feasibility of Integrating Heating Valve Drivers with KNX-standard for Performing Dynamic Hydraulic Balance in Domestic Buildings*, World Academy of Science, Engineering and Technology, Issue 49, 2011, pp. 367 - 372.
- Zell, A. (2000). *Simulation neuronaler Netze*. Oldenbourg Wissenschaftsverlag, ISBN 3-486-24350-0, Oldenbourg, Germany