Machine learning-based models for self-learning indoor heat warning systems in households

Oscar Villegas Mier[1], Willi Haag[1], Raghavakrishna Devineni[1], Guillerme Carraro Carella[1], Rainer Gasper[1], Jens Pfafferott[1], Michael Schmidt[1]

[1]Offenburg University of Applied Sciences
oscar.villegas@hs-offenburg.de

Abstract. With climate change and global rising temperatures heat health warning systems have become important in accurately predicting heat waves. However, most heat health warning systems rely on the ambient temperature forecast and do not take indoor building conditions into consideration. Moreover, a general heat warning system cannot accurately predict the heat stress conditions in individual buildings. To implement the prediction algorithms the study also proposes a Raspberry Pi based measurement system. Furthermore, to reduce the computational load on Raspberry Pi a Transfer learning technique is implemented from a pre trained Long Short-Term Memory (LSTM) neural network. The results show prediction accuracy of 97% with an RMSE of 0.218 for indoor temperature prediction.

Keywords: Self learning; Neural networks; Grey-box models; Blac-box models, heat warning, building thermal dynamycs

1 Introduction

Global temperatures are increasing due to global warming, leading to a higher occurrence of hot days with temperatures exceeding 30°C in Germany. Longer heat waves can have various impacts on the population leading to higher health risks as hot weather can be a major cause for stress in the human body [1]. A key challenge for maintaining health in heatwaves, relies in reducing heat strain via an adequate management system that ensures thermal comfort [2]. This is possible with different measures such as efficient ventilation and shading, using air conditioning systems, among others. A big challenge in the correct implementation of these measures is the assessment of stress in individual environments [3].

Timely preparation and prevention are the most effective measures against heat stress. Most warning systems in operation are restricted to the forecast of outdoor heat stress and they don´t account for specific conditions inside buildings [4]. As the heat impact of buildings indoors varies due to local conditions, each building must be rated separately. For this purpose, a system is needed that can be integrated into existing buildings to reliably measure and evaluate various building specific parameters to determine and predict building-specific conditions of heat stress, determine specific actions, and issue a heat warning.

To make this kind of system widely available, it should be compact, low power consuming and low cost. In recent years artificial intelligence (AI) capable, low cost, compact processing devices such as the Raspberry Pi have been more widely available in the market, but due to their size, their processing power is still constrained. This presents a challenge in terms of deploying and running complex algorithms to process the information and learn from the conditions of the environment where they are placed. The computational requirements of these algorithms may exceed the capabilities of the device. Furthermore, the limited memory capacity of these devices may pose constraints on the size and complexity of the models that can be deployed [5].

In this paper, we will discuss the development and implementation of self-learning algorithms developed with AI techniques to perform temperature predictions and deliver warnings within our low cost in-house developed heat warning system. The heat warning system was developed as part of the project heatGUIde, which is supported and funded by the Baden-Württemberg Stiftung. It was designed to monitor and predict extreme heat stress events in residential buildings. It consists of the integration of multiple sensors to measure thermal comfort parameters such as temperature, humidity, wind velocity, mean radiant temperature (from a black globe temperature sensor), light, and

CO2. The design is based on an Internet of Things (IoT) architecture, with a central gateway based on a Raspberry Pi computer as the main core for processing, and LoRa modules for data transfer.

2 System Design & Architecture

The heatGUIde heat warning system was built following methods based on the architecture of Internet of Things (IoT) systems for the data collection. The devices based on these architectures are often resource-constrained, small sized and can be battery-powered due to their low energy consumption [6]. This makes them a satisfactory solution for a low cost, multi sensor device. The sensors for the measuring system were selected based on cost, energy consumption, accuracy, and availability in the market.

The measuring system consists of a node and a gateway. The node measures the indoor comfort parameters and transfers this data to the gateway. The node consists of low-cost sensors hardwired to an ESP32 microcontroller enclosed in a 3D printed casing. The sensors measure Temperature (°C), and Relative Humidity (%rh), CO_2 Concentration (ppm), Wind Speed (m/sec), and Mean Radiant Temperature (°C).

The ESP32 reads out the measurement data from the sensors and sends the data wirelessly to the gateway. The gateway is the central processing unit of the measuring system. It consists of a Raspberry Pi, LCD screen and a LoRa Hat. LoRa radio communication is used to communicate between the node and the gateway. LoRa is a proprietary radio communication technique operating sub gigahertz radio frequency band capable of data transfer over longer distances and through obstacles like thick walls. The LoRa hat in the gateway makes the Raspberry Pi LoRa capable there by circumventing the use of standard LoRa gateways available on the market.

The gateway consists of necessary software for data collection, storage, analysis, visualization and the algorithms for data learning and prediction. The system architecture and the information flow through different software is shown in Figure 1.

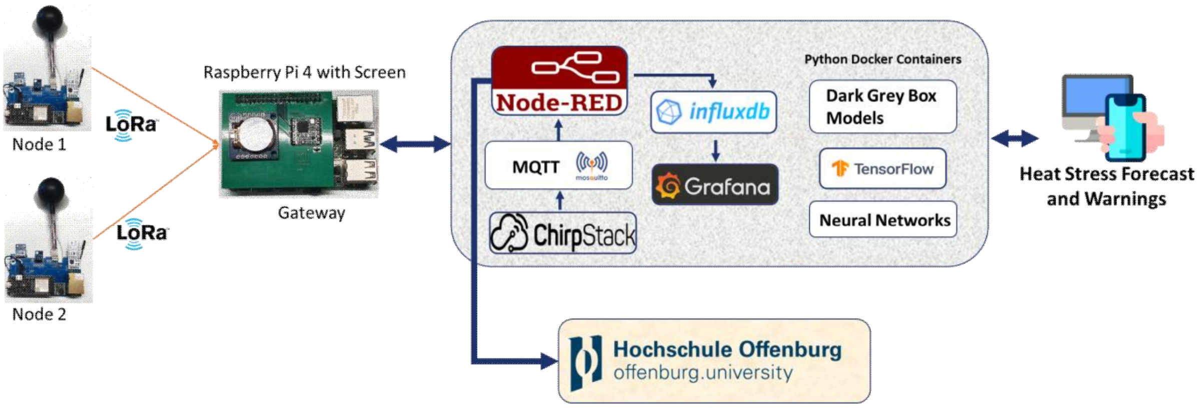

Figure 1. System architecture and data flow of the measurement system for the project heatGUIde

The data received by the gateway is decrypted by ChirpStack (LoRa WAN Server software), which is then sent to Node RED using mosquito MQTT broker. The Node RED is a graphical developing software, which directs the data from the MQTT broker to various destinations. Additionally, Node RED can also be programmed to display interactive dashboards on the gateway screen. Different models discussed in this study are deployed in different Python containers. The Node RED communicates the data with different Python containers to forecast heat stress indicators. The calculated metrics are saved in an influx Database locally and on a remote server for backup. The important results and forecasted heat warnings are displayed through interactive dashboards on the Gateway.

3 Methods

3.1 Data Acquisition

With the intention of making the AI models more generic and adaptable to different climatic conditions in Germany, the models should be fed with the local weather information. To achieve this, we considered the Test Reference Year (TRY) data with 14 parameters modelled from the weather data for the last 20 years. A Principal Component Analysis (PCA) was performed to identify the important parameters for forecasting.

The TRY data was incorporated into a real-world scenario at Project House Ulm. Project House Ulm is a single-family house, equipped with multiple sensors and data collecting systems for the purpose of studying real world conditions. For the initial analysis we created Long Short-Term Memory (LSTM) neural networks with data from Project House Ulm and the TRY data. A data model was built as follows:

Table *1.* Offline: Past Data, Actual Data, without forecast as features.

Features (inputs)	Predictions (outputs)	Horizon	Historical input	Results
Indoor Temperature	Indoor Temperature	1+ days (Tested 3 days)	5 days past	R2: 0.975 RMSE: 0.59

The forecast results are shown in Figure 2 which displays the accuracy of predicted values compared to the original dataset.

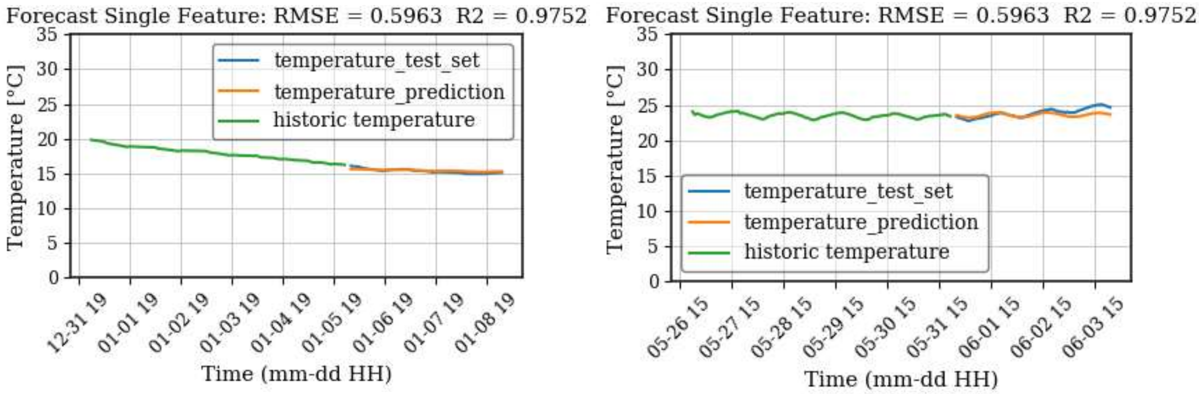

Figure *2.* Validation of preliminary analysis neural network models results. The graph shows different test set validation scenarios of temperature predictions, presenting good adaptation from the data in fluctuating temperature scenarios.

The number of features selected can impact directly in the predictions using neural networks. By means of an analysis with PCA reductions we estimated the effect on the overall prediction error. The error is relatively small when using the full 12 features in the TRY dataset. However, when reduced by up to 7 dimensions, the error remains consistently low, allowing accurate prediction with less data.

Following the initial proof of concept and PCA analysis, we shifted our focus to acquiring and utilizing real data from residential building rooms. After completing the initial proof of concept and PCA analysis, we gathered the indoor data from the heatGUIde system with outdoor data sourced from the German Weather Service (DWD), along with weather forecasts from their MOSMIX model. These combined datasets were used to train several black and grey-box models.

The heatGUIde system is integral to our data acquisition process. The data collected by these nodes was pivotal for the development, training, validation, and continuous evaluation of our diverse model approaches. The system offers real-time indoor heat stress and comfort data for model predictions and evaluations. Storing and analyzing this data helps us better understand how different environmental factors collectively affect indoor heat stress conditions.

3.2 Modeling Approach

Indoor room temperature is a principal factor involved in the physical sensation of heat, and we base our heat warning calculations mainly on it. For instance, our focus relies on accurate prediction and understanding the changes in the room temperature where our heat warning system nodes are installed.

In this work, different self-learning models were explored. To study their differences, we divided the approaches into black-box and grey-box models.

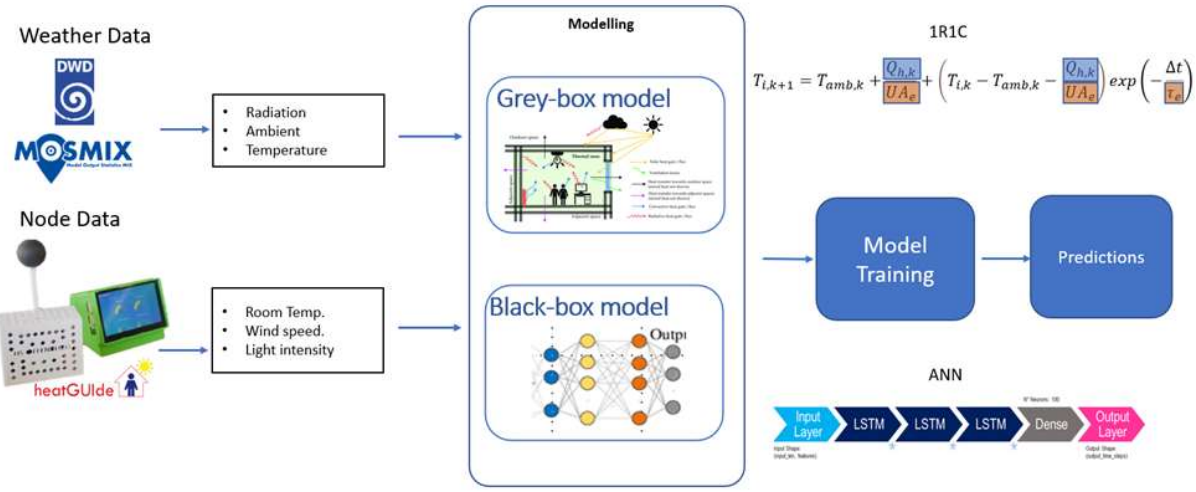

Figure 3. Data flow of the implementation of a self-learning algorithm for an indoor heat warning system

The first modeling approach explored the creation of black-box models. Black-box models are complex algorithms known for generating reliable predictions or outputs based on the self-adaptation of their input data, but with their decision-making processes being opaque [7].

Our selected black-box models were Long Short-Term Memory (LSTM) neural networks, which are a special kind of Recurrent Neural Networks (RNN), that were utilized due to their capacity to learn and remember patterns over extended sequences, crucial for time-series data inherent to temperature predictions.

In addition, we explored the use of Transfer Learning with LSTMs to reduce the processing requirements. Transfer learning is a machine learning technique that works based on using pre-trained models developed for a specific task and trained on extensive datasets and reusing the model as the starting point for a model on a second task, leveraging pre-learned patterns for enhanced learning efficiency and accuracy. It's especially useful in scenarios where the data is scarce, or computational resources are limited.

The second modeling approach uses grey-box models which are a type of predictive model that combines both theoretical and empirical approaches, offering a balance between the transparency of physical models and the predictive power of black-box models. They incorporate known physical processes with data-driven components, facilitating interpretation while maintaining predictive accuracy [8].

The selected grey-box model included in this study is a 2R2C model. An RC (Resistor-Capacitor) is a simplified representation of building thermal analysis to simulate and understand the thermal behavior and heat transfer processes within buildings. It uses electrical analogs of thermal resistances (R) and capacitances (C) to model the heat flow and thermal storage within building elements.

In comparison to a 1R1C model which is a very simple lumped parameter model, the 2R2C model introduces an additional layer of complexity and precision. While computationally more intensive, it provides detailed insights into thermal dynamics, improving the accuracy of the predictions.

4 Implementation

Each of the selected modeling approaches works with distinct methodologies which require specific tools for their correct development and implementation.

4.1 Platform and Tools

All the software frameworks on the Raspberry Pi utilize docker containerization. The model implementation, conducted using Python 3.10, is described in Figure 5, outlining the process of training and validation.

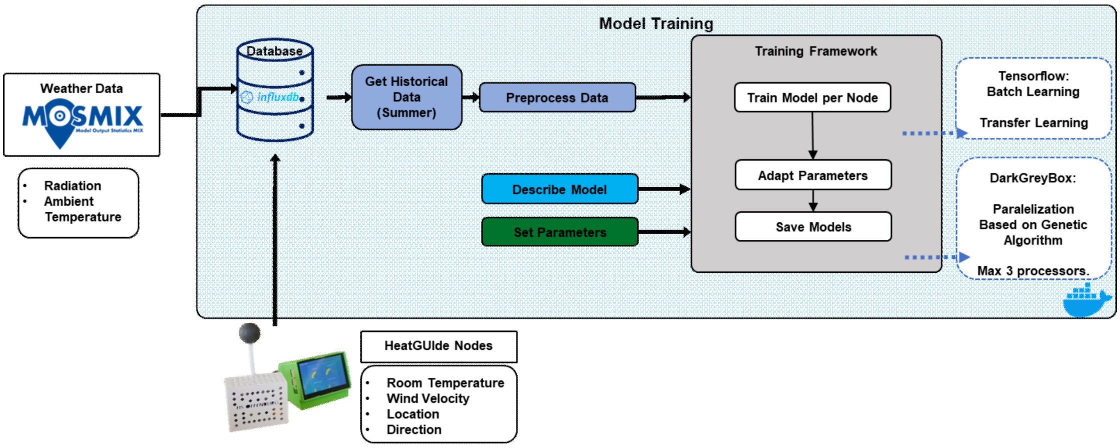

Figure *4*. Model Training implementation

Both model implementations require the use of measured data. As mentioned in Section 3, the collection of real data is important for this step. The required data is first retrieved from the database, and then preprocessed with python Pandas to fill in gaps. To facilitate the training and prediction, and to match the heat warnings, the data was resampled to one-hour intervals. As some data is measured in smaller samples, data must be up sampled, and with matching sampling rates, the datasets are joined and prepared for the modelling.

4.2 Black-box models and transfer learning.

For the black-box models, python TensorFlow and Keras were employed due to their extensive libraries, community support, and scalability offering the flexibility and resources necessary for building and training complex neural networks. This combination facilitated a smooth development process, allowing for efficient design, training, and validation of the LSTM neural networks. The availability and support of a full TensorFlow installation, containerized within the Raspberry Pi environment, were significant factors influencing the choice of this framework.

The selected features were divided into two categories, and scaled separately using min-max normalization as follows: The first feature uses the indoor temperature which was time shifted to be given as the predicted value and kept in its current state as feature. Second the general conditions of the prediction, such as ambient temperature, radiation, wind speed, and time derived features, such as the time of the day, month, weekday, weekend, and season. The latter features were scaled separately to re-scale the outputs independently when performing predictions. The final structure of the Artificial Neural Network (ANN) model is shown in Table 2. The model was trained during 400 epochs, with an early stopping callback with persistence of 20 epochs, and with a batch size of 32, using the ADAM optimizer with a mean squared error as loss function.

Table *2*. LSTM neural network architecture for black-box model.

Layer (type)	Output shape	param
Input Layer	(None, 1, 14)	Neurons: 14

LSTM	(None, 1, 100)	Neurons: 100, Return sequences: True, Function: Tanh
Dropout	(None, 1, 100)	Value: 0.9
LSTM	(None, 1, 50)	Neurons: 100, Return sequences: True, Function: Tanh
Dense	(None, 1)	Neurons: 1, Function Linear

Training neural networks on the Raspberry Pi presented significant challenges due to its limited processing power and memory capacity, therefore we explored two approaches: training with the full dataset and employing transfer learning.

Using the complete dataset provides the model with a broader understanding of the processes happening in the data, but it is computationally intensive for the Raspberry Pi. To mitigate this limitations as a second approach, models were initially trained on a more powerful system using a subset of the data, then fine-tuned on the Raspberry Pi using real-time data from the heatGUIde nodes. This approach ensured the models could efficiently learn and adapt to the unique characteristics of each household without overwhelming the Raspberry Pi's computational resources.

To implement transfer learning, we executed a simple strategy, which consists in pruning the original model from their output layer, substituting it for a new output layer, and restricting the training of the layers to the last one. This way the original layers remain so-called "Frozen" and only low computational requirement is needed to execute the adaptation of the output layer. The model was trained during 100 epochs, with batch size of 32, using the ADAM optimizer with a mean squared error as loss function.

4.3 Grey-box models.

The grey-box models were implemented using the DarkGreyBox [9] library for Python, which is specifically designed for the creation, training, and validation of grey-box models. It provides a data-driven approach within the classic Machine Learning (ML) framework for model performance evaluation. At the same time, it allows to setup and select the best performing from a series of competing models based on principles inspired by Genetic Algorithms (GA), addressing the main disadvantages of training grey-box models, that require initial condition values for the thermal parameters to be pre-calculated [10].

The primary challenge was finding a balance between model complexity and optimization efficiency for the 1R1C and 2R2C models. While a complex model could accurately describe a building's thermal dynamics, it would also be computationally demanding, making it impractical for real-time applications on low-power devices.

Careful experimentation and iteration led to models that were complex enough to accurately represent building dynamics but streamlined enough to allow for efficient optimization. This approach ensured the models were both accurate and practical for deployment on low-capacity systems like the Raspberry Pi.

The successful implementation of these methods implies that accurate heat stress prediction in households is achievable on low-cost, low-capacity systems. Each model is finely tuned for specific locations, ensuring precise and reliable predictions. These location-specific models are not only economically feasible but also efficient and effective in predicting indoor temperature, paving the way for widespread adoption and implementation in various households to safeguard residents against the health risks associated with heatwaves.

5 Results & Discussion

The proposed models were trained with data from the heatGUIde nodes which were tested and collected data in a real house in Offenburg Germany.

5.1 Black-box Model Analysis

The first approach encapsulates the training of neural networks utilizing the entire dataset within the Raspberry Pi environment. Validation results denote a notable precision in predictions. Graphical depictions demonstrate a robust correlation between the predicted and actual indoor temperatures (refer to Figure 5) showing a prediction accuracy of 97% with an RMSE of 0.218.

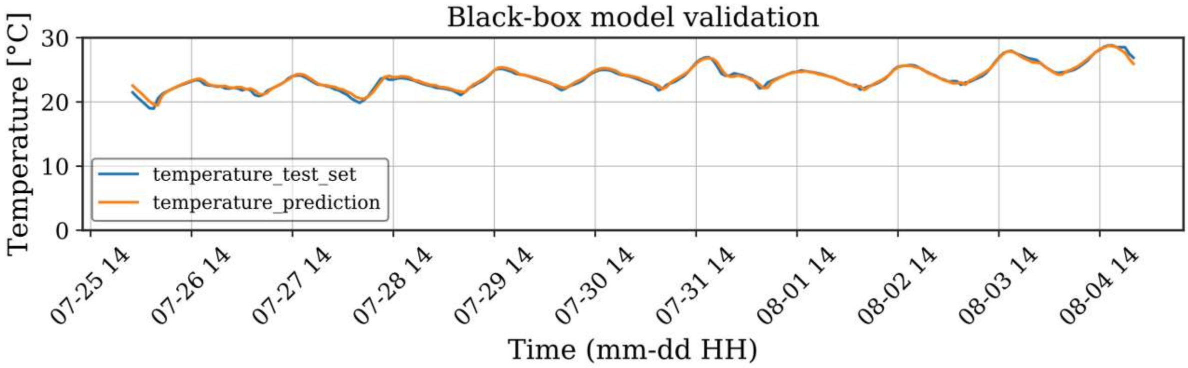

Figure *5*. Test-set validation results for LSTM neural networks, trained with batch data approach.

Despite the efficacy in predictions, there's a consequential demand on the Raspberry Pi's computational resources. Table 3 describes the CPU utilization and memory allocation statistics during the training phase. These quantitative measures underscore the substantial resource and processing power requirements of this approach, posing potential feasibility challenges for sustained, real-time applications.

Table *3*. Resource utilization for model training in raspberry Pi.

Method	Clock-Time (s)	CPU-Time (s)	Memory (GB)
Black-box (Full dataset)	169.63	231.91	1.78
Black-box (Transfer learning)	62.30	62.32	1.74
Grey-box	335.52	44.44	1.73

5.2 Transfer Learning Approach

Alternatively, models instantiated through transfer learning exhibited a rapid acclimatization to the unique dataset characteristics. Showing a prediction accuracy of 72% with an RMSE of 0.108 (refer to Figure 7).

This methodology manifested a substantial reduction in the computational resource's requirements, a pivotal consideration for the Raspberry Pi's limited hardware capabilities. The efficient training process, reduced resource requirements, and good results of this approach renders it a viable candidate for real-time predictive modeling in constrained environments.

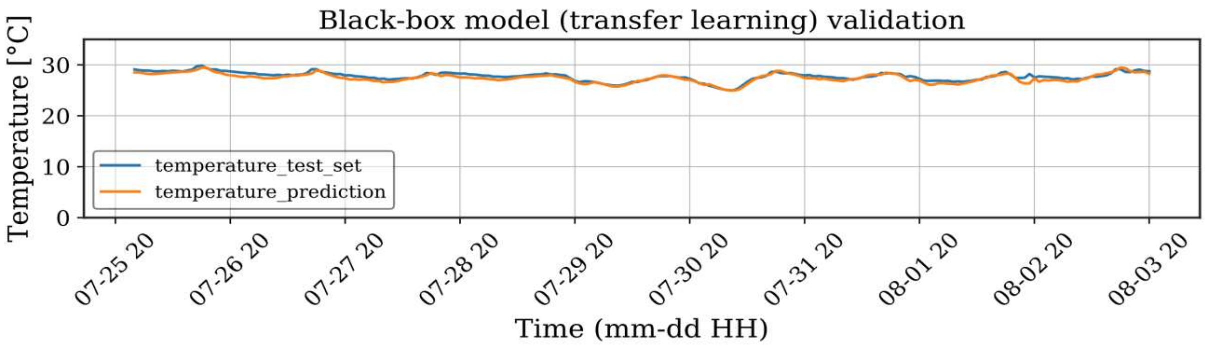

Figure *6*. Test set validation for black-box model with transfer learning approach.

5.3 Grey-box Model Analysis (2R2C Model)

The 2R2C grey-box model's implementation on the Raspberry Pi underwent rigorous validation procedures. The resulting data shown in Figure 7, illustrates a coherent and reliable fit of the data within the model. Showing a prediction accuracy of 85% with an RMSE of 1.765. Despite the complexity of the 2R2C model, the Raspberry Pi proficiently managed its optimization and training phases, with the detailed resource requirements shown in Table 3. To accommodate better resources for parallel tasks required by the programs described in Section 2, we limited the parallelization to three out of the four available processors, ensuring uninterrupted program execution. This successful implementation shows the model's viability for deployment in environments with restricted computational resources.

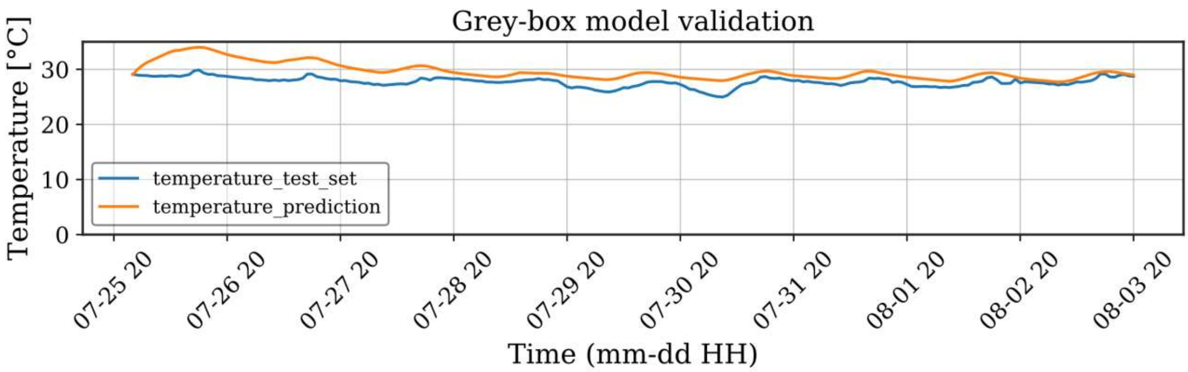

Figure *7*. Test set validation for grey-box 2R2C model with adapted parameters.

5.4 Discussion

Black-box models offer an efficient, accurate, and straightforward means to deliver predictions. At the same time, they inherently suffer from a lack of interpretative transparency. This characteristic, that can be proven advantageous for quick modelling and implementation requiring little building thermal dynamics knowledge, hampers the intuitive understanding of their decision-making, complicating potential troubleshooting and replicability. Also, in the case of our research, further implementation of control strategies can become complex due to the missing binding of physical properties, to the input variables. Although this can be overcome with the use of hybrid models, their implementation becomes more complex.

Conversely, the grey-box approach exemplified by the 2R2C model brings together predictive accuracy with an unparalleled transparency, facilitating a more intuitive understanding of the intricate decision-making processes at play. This clarity not only streamlines troubleshooting but also simplifies the model's expansion and modification endeavors. The ensuing versatility and adaptability, coupled with dependable accuracy, make the grey-box models particularly appealing for diverse and dynamic real-world applications such as control related requirements.

Through comprehensive analysis and validation, our findings demonstrate that the grey-box models can provide a confluence of predictive accuracy, model transparency, and computational efficiency. This renders this approach particularly well suited for deployment in small computational devices, such as the heatGUIde system, providing in this sense pivotal insights for the future developmental trajectory and application spectrum of household AI-driven heat warning systems.

In comparison to prevailing heat warning solutions, our heatGUIde system introduces several distinguishing features that substantially elevate its value proposition for end-users seeking efficient household heat reduction solutions such as low-cost, and the inclusion of Self-Learning Predictive Models. These models give the system dynamic adaptability by providing personalized predictions of the room conditions, which can aid better heat stress reduction measurements.

The use of the self-learning models is not only to predict the room conditions but can also be used use analyze and recommend specific actions users might take to preemptively mitigate heat stress, going beyond simple warning systems to act as an advisory companion for users in heat management.

Future work will focus on improving the learning capabilities of the system, to continually refine and enhance its predictive accuracy over time. By learning from new measured data, the system becomes progressively more adept at forecasting indoor temperatures and heat stress scenarios, ensuring that its utility improves the longer it is in use. Additionally, focus on user experience is previewed to enhance its adoption and foster seamless integration into households. Lastly as the system is designed with future integrations in mind. Its architecture and algorithms are being designed for compatibility with smart home systems. For instance, it could dynamically adjust smart blinds or ventilation systems in anticipation of changing indoor temperatures, providing not just a reactive but a preventative solution to indoor heat stress.

This advanced, yet accessible system thus stands as a significant contribution to the field, with the potential to drive further innovations and improvements in the realm of household heat management and control technologies. Our research encapsulates the next stride in indoor environmental management, offering a glimpse into the future of smart, safe, and comfortable living spaces.

6 Conclusion

Through a structured and phased data acquisition strategy, we ensured a systematic approach to understanding and analyzing indoor thermal dynamics. The initial phase provided a theoretical and practical understanding, while the usage of the data collected from the heatGUIde system, offered real-world, applicable insights, collectively laying a solid foundation for our subsequent methodological applications and algorithm developments. This diverse dataset was indispensable for the development and fine-tuning of our models, providing a rich and varied source of information to train, test, and validate our algorithms effectively and efficiently.

Considering the escalating global temperatures and the increasing prevalence of heatwaves, the successful implementation of these methods implies that accurate heat stress prediction in households is achievable on low-cost and low-capacity systems. Each model is finely tuned for specific locations, ensuring precise and reliable predictions. These location-specific models are not only economically feasible but also efficient and effective in predicting indoor temperature, paving the way for widespread adoption and implementation in various households to safeguard residents against the health risks associated with heatwaves.

Within this research we also introduced and elaborated on the heatGUIde system, a novel, user-centric solution designed meticulously for effective and efficient indoor heat stress mitigation in households. The system distinguishes itself by melding affordability, intuitive design, proactive advisory features, and the integration of self-learning algorithms. Each attribute is not standalone but intricately interwoven to construct a holistic solution that actively navigates and manages indoor environments to safeguard occupants during heatwaves.

7 Acknowledgments

We gratefully acknowledge financial support by Baden-Württemberg Stiftung through project heatGUIde.

8 References

[1] Umweltbundesamt, *Effects of climate change clearly noticeable in Germany.* [Online]. Available: https://www.umweltbundesamt.de/en/press/pressinformation/effects-of-climate-change-clearly-noticeable-in (accessed: Oct. 12 2023).

[2] F. S. Arsad, R. Hod, N. Ahmad, M. Baharom, and M. H. Ja'afar, “Assessment of indoor thermal comfort temperature and related behavioural adaptations: a systematic review,” *Environ Sci Pollut Res*, vol. 30, no. 29, pp. 73137–73149, 2023, doi: 10.1007/s11356-023-27089-9.
[3] K. Lundgren Kownacki, C. Gao, K. Kuklane, and A. Wierzbicka, “Heat Stress in Indoor Environments of Scandinavian Urban Areas: A Literature Review,” *International journal of environmental research and public health*, vol. 16, no. 4, 2019, doi: 10.3390/ijerph16040560.
[4] J. Pfafferott, S. Rißmann, G. Halbig, F. Schröder, and S. Saad, “Towards a Generic Residential Building Model for Heat-Health Warning Systems,” *International journal of environmental research and public health*, vol. 18, no. 24, 2021, doi: 10.3390/ijerph182413050.
[5] M. Zhang *et al.,* “Deep Learning in the Era of Edge Computing: Challenges and Opportunities,” in *Fog Computing*, A. Zomaya, A. Abbas, and S. Khan, Eds.: Wiley, 2020, pp. 67–78.
[6] G.C. Madhu, P. Vijayakumar, and X.Z. Gao, “Resource constrained IOT environments: A survey,” *Journal of Advanced Research in Dynamical and Control Systems*, vol. 9, no. 16, pp. 445–457, 2017. [Online]. Available: https://www.researchgate.net/publication/329310950_Resource_constrained_IOT_environments_A_survey
[7] J. Yu, W.-S. Chang, and Y. Dong, “Building Energy Prediction Models and Related Uncertainties: A Review,” *Buildings*, vol. 12, no. 8, p. 1284, 2022, doi: 10.3390/buildings12081284.
[8] Y. Li, Z. O'Neill, L. Zhang, J. Chen, P. Im, and J. DeGraw, “Grey-box modeling and application for building energy simulations - A critical review,” *Renewable and Sustainable Energy Reviews*, vol. 146, p. 111174, 2021, doi: 10.1016/j.rser.2021.111174.
[9] *GitHub - czagoni/darkgreybox: DarkGreyBox: An open-source data-driven python building thermal model inspired by Genetic Algorithms and Machine Learning.* [Online]. Available: https://github.com/czagoni/darkgreybox/tree/master (accessed: Oct. 13 2023).
[10] Z. Shi, G. Newsham, A. Pardasani, and H. B. Gunay, “On Formulation and Training of Grey-box Thermal Model for Low-rise Residential Buildings,” in *Proceedings of Building Simulation 2019: 16th Conference of IBPSA*, Rome, Italy, 2020, pp. 838–844. Accessed: Dec. 10 2023. [Online]. Available: http://www.ibpsa.org/proceedings/BS2019/BS2019_210251.pdf