Machine Learning-based Ambient Temperature Prediction in Radio Access Network Environments

Selma Rahman [1*], Mattias Olausson [1*], Carlo Vitucci [1,2*] and Ioannis Avgouleas [1*]

[1*] Ericsson AB, Stockholm, Sweden .
[2*] Mälardalens University, Västerås, Sweden .

*Corresponding author(s). E-mail(s): Selma.Rahman@ericsson.com; Mattias.Olausson@ericsson.com; Carlo.Vitucci@mdu.se; Ioannis.Avgouleas@ericsson.com;

Abstract

Machine learning is revolutionizing various fields, but its implementation in real-time soft environments often faces challenges due to limited computational and storage resources. In this work, we have successfully developed a highly accurate Random Forest regression model to predict the working ambient temperature for an embedded Radio Access Network system, particularly within the Baseband application domain. Our model achieves minimal prediction error and maintains a variance well-aligned with the onboard sensors' measurement accuracy. Remarkably, the outcomes of our research respect the stringent real-time processing and storage constraints, making it a significant advancement in real-time machine learning applications.

Keywords: Predictive Maintenance, Temperature prediction, Radio Access Network

1 Introduction

The number of publications in the Artificial Intelligence and Machine Learning (AI/ML) domain has tripled in the last ten years (see Figure 1). Attention to the research field knows no respite, and the growth in AI patents is exponential (see Figure 2). It could not be otherwise: one of the discriminants for using AI/ML effectively is the availability of data, an event that the continuous shift of the network towards a cloud-oriented architecture has made possible.

The availability of large amounts of data has made the use of AI/ML increasingly attractive and essential, as well as the means to analyze this enormous amount of data and identify helpful information patterns. Consequently, machine learning dominates the number of AI publications without rivals. Another fundamental pillar for efficiently applying machine learning is the application domain knowledge (see Figure 3). Publications spread through many domains, from medical to networking, automotive, and manufacturers.

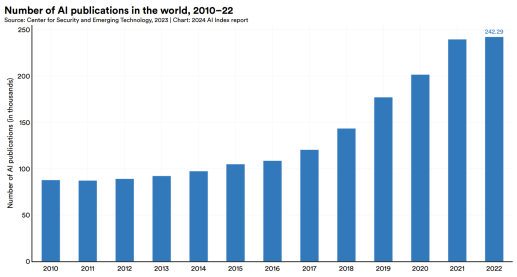

Fig. 1 AI number of publications trend 2010-2022 [1]

Each domain has its characterization and limits that are discriminant in implementing machine learning. The Radio Access Network (RAN) is no exception and is the domain of our interest. The RAN characterizations that influence the development of machine learning are limited availability of resources, computational, memory, or networking, and a "temporal" limitation, in the sense that the system is sensitive to disturbances typical of soft real-time that can lead to adverse effects on performance and, in the most extreme cases, malfunction.

The development of fifth-generation telecommunications, the so-called 5G, was not driven by technological evolution but by a commercial necessity. In fact, with the advent of smartphones, the value of the network has progressively shifted from connectivity to the data. 5G represents the opportunity for the operators to enter the rich market of services, making their business model and investment in network infrastructure sustainable. The core business shifts from connectivity to service deployment, and operators can generate profits by hosting a broad set of services in their infrastructure, close to the end user. However, 5G has led to increased infrastructure complexity due to:

- increased throughput and delay requirements [2],
- widespread computing capacity deployment (especially for dense urban areas) [3], and
- intelligent self-monitoring and easily-maintained configuration system to decrease Capital Expenditures (CAPEX) and Operational Expenditures (OPEX) [4].

Once the value of the network move to services, the network needs a high resiliency level to guarantee high quality of experience from end user to be a sustainable new business case. If the resilience is the ability to the system to react to a disturbance and recover the requested functional ratio condition, the fault management is the system function to achieve the resilience goal [5]. Consequently, the need for a fault management framework that is strongly oriented towards the centrality of the recovery action has also grown in tandem with the complexity of the infrastructure: the fault management now aims to detect, locate, recover and predict a fault condition [6] because fault prediction [7, 8] and predictive maintenance [9] derive from the need of increasing the infrastructure sustainability.

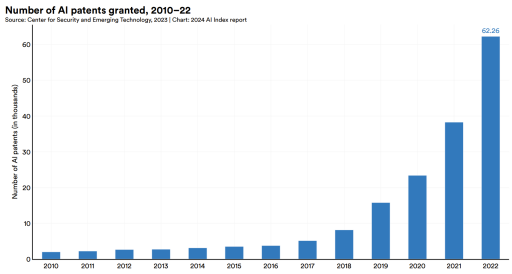

Fig. 2 AI number of patents trend 2010-2022 [1]

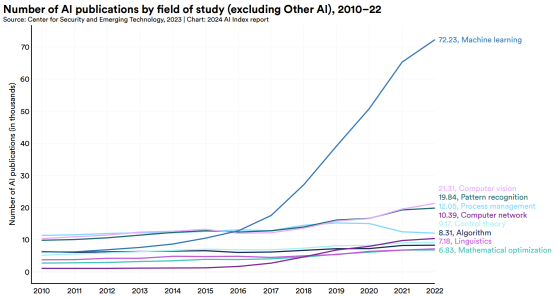

Fig. 3 The weight of the AI subset functions in the total number of publications [1].

1.1 Context Description

Our research focuses on the ability to do predictive maintenance for products in the Radio Access Network (RAN) domain. The "cloudification" of the network suggests a technological convergence with data center hardware products, but the environmental conditions are very different. A RAN solution, for example, must rely on something other than the cooling systems available for data centers due to cost, space, and noise constraints. Furthermore, RAN products should work under very different circumstances, e.g., their operating temperature spans a more demanding range than the typical temperature for data center products.

The aforementioned scenario highlights that the validity of research findings concerning the correlation between environmental parameters and system reliability is confined to the specific domain of reference. Another characteristic of RAN products is that they poorly tolerate disturbances and interruptions.

The data acquisition process must be unique regarding environmental and work parameters,

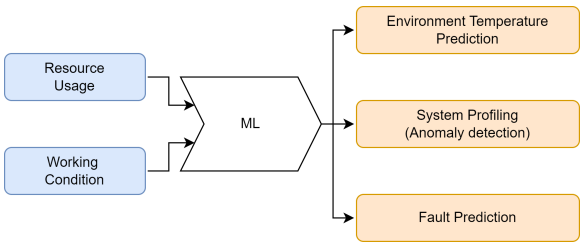

Fig. 4 The data analysis and pattern finding in baseband boards.

i.e., the use of system resources. Furthermore, data collection is crucial for network access systems since they are often called for hosting soft real-time systems. The latter exhibit stringent requirements in terms of the reaction time and execution of a particular task such as the reception and decoding of traffic packets. The collection of data must therefore be as least intrusive as possible so as not to compromise the functionality of the node and the availability of bandwidth when transmitting the collected data.

Figure 4 shows the machine learning usage in the baseband board. Resource usage and environmental conditions are the data flow feeding the machine learning algorithms. Based on the model, fault management can use fault prediction to enrich the fault location, improve fault detection, and reduce the recovery time. It is possible to identify three main outocomes:

- Detect malicious or malfunction condition in the environment setting: the environment temperature prediction.
- Detect malicious or malfunction condition in execution flow: the system anomaly detection.
- Detect malicious or malfunction condition in the baseband components: the fault prediction.

The more distributed computing and high data traffic capacity also involve a considerable workload. The evolution of hardware design on the nanoscale has been the response to this growth in data processing for both the latest generation processors and memory devices (DDR5). The reliability of hardware components has indeed increased in recent years [10], but it is equally valid that the complexity of the design has also increased. And, with the nanoscale hardware design, the probability of temporary or permanent fault conditions is higher due to power fluctuations, excessive operating temperatures, or cosmic radiation. Eventually, the hardware will end its life due to aging issues, and the system reliability will enter a critical phase where the failure rate will increase exponentially.

The hardware repair process is costly: maintenance activities on-site, packaging, transportation, board troubleshooting, and test to confirm the failure condition diagnosis for the component, and faulty hardware replacement, if applicable. In telecommunication networks, multi-chip packages, robotics, automotive, and, more generally speaking, in an increasingly widespread distributed system, the hardware devices must work and inter-work properly, react to external disturbances promptly, and operate as long as possible. However, it must use an appropriate error prediction action by analyzing the data available from the system. Without this fundamental prediction action, the maintenance costs could be relatively high. Thus, it is essential to know how to identify a possible failure condition before it happens. Understanding how the state and use of resources affect their life cycle allows planning appropriate recovery actions in time, whether an actual replacement of the component or preventive isolation to enable an operational state in full or degraded function mode. Predicting the hardware fault is, therefore, fundamental for the sustainability of the future network. Without it, the unsustainable maintenance cost would compromise developing innovative services for industry 5.0 [11]. Machine learning and Artificial Intelligence can be the technology enabler for a fault prediction based on system data [12].

1.2 Problem Statement

The probability of device failure, and thus its lifecycle, is significantly influenced by the ambient temperature [13]. Clear indications of ambient temperature enable the implementation of optimal preventive measures, such as controlled activity cooling (thermal throttling), dynamic control of the cooling system, and continuous reassessment of the product's lifecycle to support a planned maintenance strategy. However, in RAN environments, baseband boards can operate under widely varying ambient conditions depending on the installation site. Accurately predicting the

ambient temperature becomes essential to signal critical conditions. For instance, an anomaly in ambient temperature might indicate an issue with the cooling system or an unexpected board temperature, which could suggest suspicious heating of one or more hardware components on the board.

1.3 Research Objective

The paper assumption is that the likelihood of a system error depends on the environmental parameters, like temperature and humidity. Those environment parameters drive the entire life cycle of the hardware devices: board working continuously under stressful environment condition will have a shorter lifetime. Our research objective is to devise a model capable of predicting the ambient temperature of the board, i.e., the temperature of the immediate surroundings of the board. The latter has a direct impact on the board's operating temperature so an accurate ambient temperature model will allow for:

- implementing operations e.g., thermal throttling, that maintain the temperature of the device below a critical threshold, and
- forecasting the component's life cycle according to the ambient temperature for optimal maintenance planning.

1.4 Research Methodology

The paper is a quantitative empirical studies [14] that aims to examine the relationships between environment parameters and resource usage through a machine learning approach. For the evaluation of temperature prediction algorithms, the research used two types of data: environmental (i.e.: temperature and humidity) and resource use (number of cores used and their load). The data refer exclusively to industrial baseband boards, and this paper used them in respect of a confidential agreement. We have also used a thermal chamber to simulate different temperature working environments. We have verified the temperature prediction algorithms' validity by comparing them with other solutions proposed in the literature. Baseband board designers have reviewed the research outcomes and evaluated implementation feasibility and sustainability in the RAN domain. With this approach, the advantage for the industrial partner is the possibility of reducing OPEX and the maintenance cost in the next generation of telecommunications systems.

Building upon the consolidated results presented in [15], our research methodology involved a comparative analysis of machine learning algorithms. Specifically, we focused on evaluating the performance of the Random Forest algorithm and XGBoost across various experimental scenarios, with an emphasis on optimizing the Random Forest approach because Random Forest algorithm outperforms XGBoost in all experimental cases in [15].

We paid close attention to the error margin of temperature sensors (+/- one degree), which was crucial for our analysis. To make our temperature measurements easier to work with, we converted the continuous range of real numbers to a discrete set of integers by using a rounding approximation mechanism (see 3.1 for more details). We also improved the quality of our dataset by implementing a procedure to check and clean any missing values. Both of these steps significantly contributed to the accuracy of the model we developed.

In the baseband environment, resource availability is minimal. The use of local storage resources, computational resources, and bandwidth, both internal bus and system backhaul, is conditioned by the need not to affect operation. To limit the performance impacts, we have reviewed and corrected the number of attributes and the maximum operating range for temperature with careful consideration. This decision is aimed at limiting data streaming and collection time, ensuring the efficiency of our operations.

The listed improvements allow us to focus on efficient analysis of the temperature prediction algorithms and their potential impact on industrial baseband board design. The refined data handling and streamlined methodology contribute to a more targeted and expedited research process. These adjustments enhance the credibility and relevance of the study's findings, offering valuable insights for optimizing resource usage in the context of evolving telecommunications systems.

This paper must comply with NDA regulations and restrictions. The dataset and the implemented algorithm cannot be published.

2 Related Work

The ability to have a thermal model for any system is a well-known need because it is clear that, as the operating temperature increases, the reliability of the CMOS-based ICs decreases exponentially [13]. Yang et al. [16], for example, provides an interesting analysis of all those factors that negatively influence both the aging and the reliability of electronic components, such as the effects of voltage (Hot carrier injection) and temperature (Bias Temperature instability). Even considering the system as a non-divisible entity, the system's failure rate doubles for every ten Celsius degrees increase above twenty-one Celsius degrees [17]. Research on the thermal model mainly focuses on two types of algorithms [18]: those based on the thermodynamic laws and the physical characteristics of the components to find a thermodynamic model of the device [19–21] and those which, recognizing the limited capacity of a thermodynamic physical model to be representative for different types of installations, prefer algorithms that have data-driven solutions [22, 23]. The latter has received more attention from researchers recently, especially concerning the progress of AI/ML as a mechanism for evaluating predictive models. AI/ML methods have stood the test of time concerning temperature prediction by providing very accurate models for applications such as weather forecasting and temperature control in industrial environments, among others. For example, Ma et al. study demonstrates a spatiotemporal correlation for fault prediction algorithms using graph convolutional recurrent neural networks (GCRNN), which seems promising to replicate beyond the meteorological domain. In the networking domain, only a few researchers have dealt with temperature prediction in the RAN domain. On the contrary, most research works considered temperature prediction in data centers and High-Performance Computers (HPC). Therein, temperature prediction allows the intelligent implementation of energy saving utilizing workload management [24, 25], effective heat dissipation [26], and improved cooling efficiency [27]. Previous works considered the operational data of the board, such as the number of cycles per CPU or the cache metrics, and the physical characteristics of the system, such as the number of CPUs, the size and type of memory or traffic devices [23, 28]. One of the used algorithms is the long short-term memory-based temperature prediction (LSTM), an improved version of the more traditional recurrent neural network (RNN), more suitable for solving time series prediction problems. In the most significant works that have used LSTM, we point out the work of Cheng et al. [29] in the multicore and Network on Chip (NoC) domain. Neural networks are computationally demanding, and our research focuses on temperature prediction through less complex algorithms and less costly solutions to meet the requirements described in the context description section. There is an inevitable divergence in the research results we have considered. XGBoost is the algorithm frequently used in applied machine learning for structured data due to its fast speed compared with other gradient-boosting implementations [30].

3 Temperature Prediction Process

3.1 Design description

This chapter presents the design description of a machine-learning model that predicts ambient temperature, i.e., the target value based on lab measurements. We train the model using board temperature, rail, and board power sensors as independent variables while controlling computing load, environment humidity, and fan speed to simulate different board operating conditions. We evaluated XGBoost Regressor (XGB) [30] and Random Forest Regressor (RF) [31] (with and without cross-validation [32]) models to determine the most suitable for the RAN domain. We performed hyperparameter optimization for both the tree-based models to fine-tune their performance and promote better generalization. By searching for the optimal hyperparameter values, our approach is to effectively regularize the models to mitigate the risk of overfitting and enhance their ability to generalize to unseen data. We placed the RAN boards inside a climate chamber in the lab. The climate chamber allows the simulation of all possible humidities and temperature levels that the baseband is likely to encounter in the field. We collected data for different computing loads by simulating no network traffic,

minimal activity, or peak traffic conditions. Since the baseband board is a multiprocessor system, we have modified the active processing units' number and computing load to simulate different working conditions. Additionally, to simulate the environmental conditions of the installation site on the baseband board, we varied the fan speed of the cooling system. Following the well-established ML principles, we split the data into two distinct data sets:

- the **training set** that is used to train the ML model. The input features include temperature sensors, watts and power levels measured at different points of the baseband board, the relative humidity and the ambient temperature of the climate chamber, among others, and
- the **test set** that is used to assess the model's performance.

The training set is assigned a splitting ratio of 80%, while the test set receives 20%. Consequently, the collected data sets encompass the distinctive patterns that characterize the baseband board in various environmental and radio traffic conditions. We trained the ML model using the training data set to create an accurate and scalable model, making it possible to use the model for future versions of RAN boards without compromising its validity. Our evaluation metric regarding which ML model to use for environmental temperature prediction is based on the mean absolute error (MAE) i.e., the absolute value of the difference between the predictions and the targets, and R-squared (R^2). Residual analysis between the predicted and the measured ambient temperatures is considered as well.

After following the research methodology outlined in section 1.4, we enhanced the data handling by discretizing the temperature (rounding values to the nearest integer) and removing measurements with missing values for specific features. These improvements have enhanced the dataset, making the machine-learning model more robust and reliable.

The new data handling allowed us to focus on improving machine learning accuracy. However, baseband products are susceptible to resource usage in runtime and require particular attention during machine learning model deployment and implementation. The cost of the coding language to the resource usage is well-known [33], encouraging us to check the coding language for the model in the baseband product. This strategic approach aims to optimize the compatibility and integration of the machine-learning model with the operational environment of the basebands, ultimately enhancing its effectiveness and practical applicability within real-world settings. By aligning the model's coding language with the language used by the basebands, seamless integration and efficient performance can be achieved, paving the way for increased utility and impact within the RAN domain.

3.2 Execution

Table 1 The distribution of the dataset, for each setting of the controlled variables

Variable	Value	Distribution [X/Total]
DSP	Low	9/18
	Mid	7/18
	High	2/18
CPU load [%]	0	1/18
	20	1/18
	30	8/18
	100	8/18
Fan speed [%]	30	2/18
	40	1/18
	50	1/18
	70	4/18
	100	10/18
Relative Humidity [%]	0	8/18
	20-80	2/18
	30-80	8/18
Temperature Ranges [°C]	0-35	8/18
	20-55	8/18
	50-60	2/18

As described in the previous section, we continuously test the baseband in the climate chamber. Thus, the training runs with a new data set after each successful run. The current training for the ML models contains 18 datasets, each collected from their respective laboratory tests. Table 1 shows the data distribution of the various combinations of the controlled variables (DSP, fan speed, CPU load, relative humidity, and ambient temperature). For example, out of eighteen datasets, nine have DSP set to "Low", seven have DSP set to "Mid", and two have DSP set to "High", etc.

The data collected is then explored and handled appropriately for the models to process.

For the training of the models, we randomly divided the whole dataset into a training and testing set using the train-test split-function in Python (*train_test_split*()[1]) by specifying the splitting ratio to be $80 - 20\%$ respectively. The purpose of the testing set is to assess and evaluate the performance of the trained model by comparing the model's predictions with the actual values from the testing set. The performance evaluation described above allows us to measure metrics such as accuracy and residuals, which provide insights into how well the model generalizes to unseen data and, thus, performs in real-world scenarios. For the sake of presentation and to provide an efficient way to compare the predicted with the measured ambient temperature values side by side, we decided to introduce a data set referred to as unseen data. The unseen dataset contains a continuous baseband run in the climate chamber i.e. with the temperatures increasing with every measurement and it is completely excluded from the training and testing phase of the ML models. The data from the features (all variables except the target variable) is then used as an input to the models to acquire their predictions. This allows us to further evaluate the models' predictive ability of new and unseen data.

In the next stages of this project, the variable "relative humidity" has been disregarded as a driving parameter, leaving the temperature sensors as the drivers. The handling of missing values was also changed from linear interpolation to deletion of all missing values of relevant features, in other words the entire row of the dataset where there were missing values, of the features that are of interest - Power and voltage missing values were disregarded since the parameter themselves were excluded, as previously mentioned. Furthermore, the numerical values were rounded up to the nearest digit. We might have a few more test runs to add in the dataset

Data handling optimization process

We have contextualized resource usage in the specific application domain of RAN. The maximum number of resources evaluated under extreme use conditions depends on the number of end users managed by the product. Considering the distribution of traffic capacity by nodes, we have defined new limits for the dataset attribute ranges. Note that the new approach means sampling attributes in the actual use of the final product, and the dataset becomes representative of realistic use cases, even in extreme environmental conditions. This change reassures us that our data accurately reflects real-world scenarios, instilling confidence in our work.

3.3 Results

Our previous work [15] tested the baseband unit under evaluation by setting the CPU and fan speed to their maximum value (100%). Fig. 5 and Fig. 6 show the resulting prediction outcomes for this unseen data. Fig. 7 and Fig. 8 show the residual, showcasing the performance of the Random Forest Regressor (with and without cross-validation) and the XGBoost Regressor, respectively. The blue graph in the prediction figures represents the measured ambient temperature values obtained from a sensor during laboratory tests, which serves as the model's target value for accurate prediction. We opted not to apply cross-validation for the XGBoost regressor due to its generally strong performance with smaller datasets while recognizing that Random Forest could benefit from cross-validation. It is worth noting, however, that the Random Forest regressor with and without cross-validation yielded similar results, indicating that the application of cross-validation did not significantly impact its performance. In addition, it is essential to recall that in our previous study [15], the conclusion drawn remains relevant to the current research. The permutation and feature importance analyses indicated that our choice of features was adequate. The figures from the previous study showed that the temperature sensors were the primary contributors to the model's performance and prediction accuracy. At the same time, it is possible to exclude the power and voltage readings without any loss of prediction accuracy. The result shows we do not need further investigation into the XGBoost Regressor as the Random Forest model performs well enough with lower overall residuals, even with a few more kinks in the predictions.

[1] sklearn.model_selection.train_test_split, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

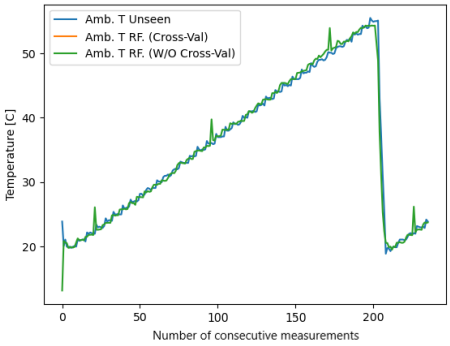

Fig. 5 Ambient Temperature Predictions, CPU=100% and fan=100%, RF Prediction

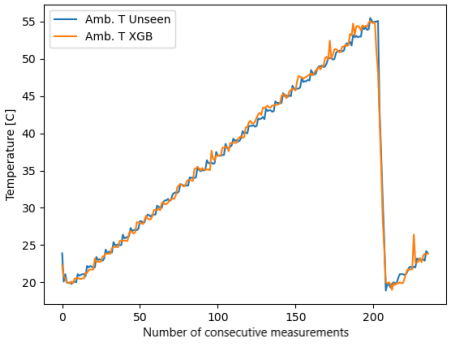

Fig. 6 Ambient Temperature Predictions, CPU=100% and fan=100%, XGB Prediction

A well-performing model should exhibit residuals, i.e., the difference between the measured (actual) value and the predicted value, scattered randomly around the horizontal line at zero on the y-axis, with no apparent patterns or trends. The absence of patterns or trends indicate that the model effectively captures the relationship between the features and the target variable and that there is no further information that it could employ to enhance its predictions. On the other hand, if the residual plot displays patterns or trends, such as a U-shape or a curve, the model fails to satisfactorily capture the underlying relationships between the features and the target variable.

Including additional information could improve the models' predictions avoiding underfitting or overfitting. Underfitting occurs when a model or algorithm fails to capture the underlying trend of the data, resulting in poor performance on training and testing data. Underfitting occurs when the training dataset is too small, the model needs to be more complex, or the data needs

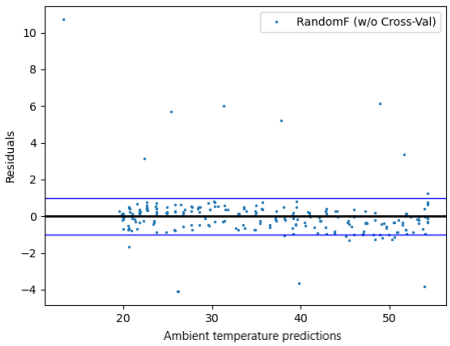

Fig. 7 Scatter plot of residuals between predictions and the measured value for a baseband with CPU=100%, fan=100%, and $\pm 1°C$ threshold displayed, RF

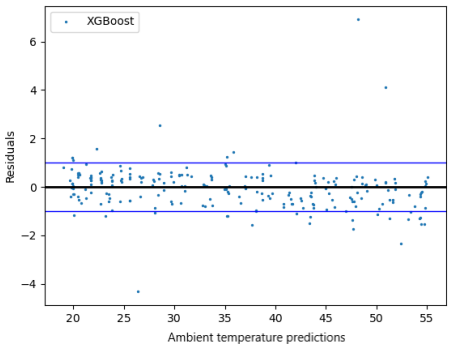

Fig. 8 Scatter plot of residuals between predictions and the measured value for a baseband with CPU=100%, fan=100%, and $\pm 1°C$ threshold displayed, XGB

to be more precise. Overfitting happens when a model is too complex and learns from noise or inaccurate data entries in the training set, leading to poor performance on testing data. An over-fitted model indicates the need to explore the reduction of the model complexity, use early stopping during training, or implement regularization, among others. Upon observing the graphs in Fig. 7 and Fig. 8 along with the graphs in Fig. 5 and Fig. 6, it is evident that the Random Forest and XGBoost regressors are capable of making predictions with a high degree of accuracy, without under- or overfitting and exhibiting errors between the range of $\pm 1°C$.

To further evaluate the accuracy of the predictions, we calculated and compared the mean absolute error (MAE) and R-squared (R^2) between the model's prediction and the measured ambient temperature of either the testing or the unseen set. These metrics provide insight into how well the

model is performing and how much of the variation in the data can be explained by the model. For instance, a low MAE suggests that the average difference between the predicted and actual values is small. In contrast, a high R^2 value indicates that the model explains a large proportion of the variance in the target variable - and vice versa. Table 2 shows the result. The models are trained successfully with relatively low error and high accuracy based on the metrics' values for the testing data, suggesting that the model fits the test data well and can make reliable predictions. Moving on to the metrics for the unseen data, it suggests that the model can generalize well and make accurate predictions on data that it has not seen before. The fact that the MAE value is lower for the unseen data than the test data suggests that the model has not overfitted to the testing data and is not capturing noise or irrelevant information. In general, these metrics indicate that the model has high accuracy and can be considered a reliable model for predicting ambient temperature.

3.4 Predictions after updated data handling

In the figures below, the ambient temperature predictions using the Random Forest model are presented, the test case CPU=100% and Fan=100% and only the temperature range 20-55 degrees to exclude the sharp decrease, when the temperatures have been rounded to the nearest integer and no rounding along with the respective residuals.

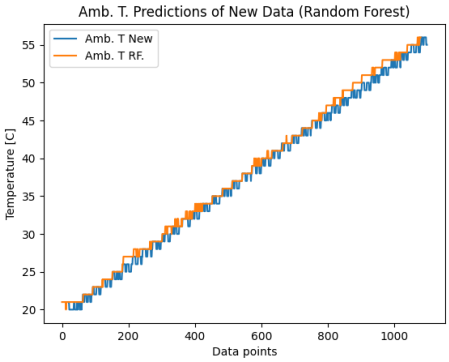

Fig. 9 Temperatures are rounded to nearest integer

In Table 2, we present the calculated mean absolute error and the value R^2 for the two cases

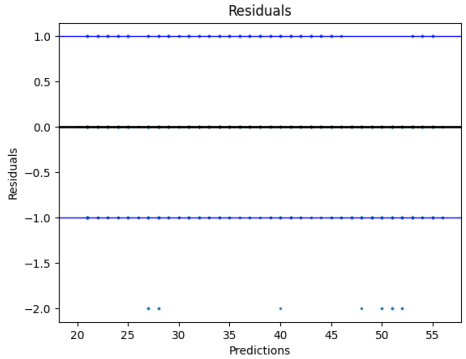

Fig. 10 Scatter plot of residuals when temperatures are rounded to nearest integer.

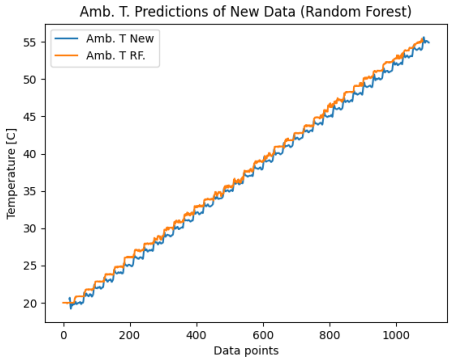

Fig. 11 Temperatures presented in non integer values

above, together with the resulting metrics from the previous study ("Prev.").

It is important to note that "New" indicates no rounding up, but the new data handling is implemented. While "Pred. Int." means only the answers, i.e., the predictions are discretized. Finally, "Int." refers to the training and unseen data, as well as the predictions, being rounded up to the nearest integer.

Table 2 MAE and R^2 values for different models when predicting baseband ambient temperature at CPU=100% and Fan=100%

Metric	**RF Prev.**	**RF New**	**RF Pred. Int.**	**RF Int.**
Test MAE	0.795	0.916	1.033	1.054
Test R^2	0.984	0.982	0.981	0.981
Unseen MAE	0.654	0.556	0.717	0.770
Unseen R^2	0.987	0.986	0.986	0.987

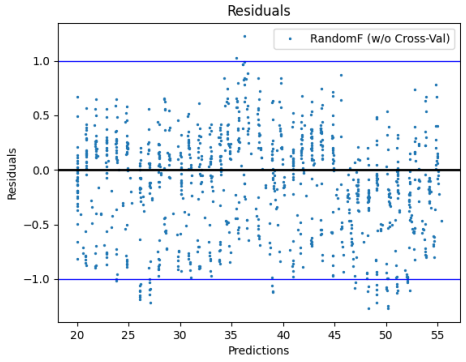

Fig. 12 Scatter plot of residuals when temperatures are not rounded to nearest integer.

Although the rounding up to integer approach yields slightly inferior results, which can be observed when comparing the Figures 9 and 10 with 11 and 12, it is deemed preferable to present temperatures as integer values because of their intuitive interpretation. In addition, using integer values simplifies the data handling process, ensuring consistency in the number of decimals across all data points.

Our research indicates that the discretization process should follow the completion of all calculations, even though the resulting MAE and R^2 values seen in Table 2 fall within runtime variance. The input data should remain original, and the process should round only the resulting predictions to integers. It is important to note that rounding input data to integers can decrease performance metrics due to error propagation each time a rounding up occurs. While this may be self-explanatory, it was necessary to investigate as our goal is to present predictions in integer format.

3.5 Predictions on under-represented training data

To assess the performance of our trained model on data that is under-represented we tested our models' predictions on a dataset for which the unseen data are: CPU load = 30%, fan speed = 70%, DSP = Low, ambient temperature range = $0 - 35°C$ and relative humidity range = 0%. The predictions can be seen in Figures. 13, 14 and 15. Insufficient dataset refers to a situation where the prediction test case lacks adequate representation in the training dataset concerning the parameter settings. It was also interesting to investigate the

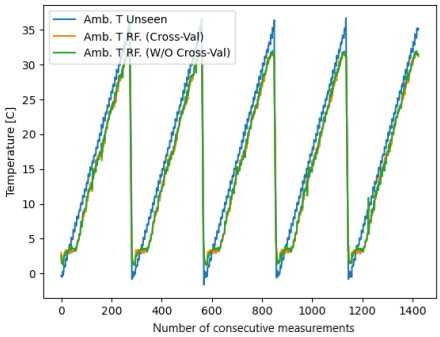

Fig. 13 Ambient Temperature Predictions, CPU=30% and Fan=70%, interpolation of MV

effect of the updated data handling on insufficient data. Figures 14 and 15 show the effect of deleting missing values from the data set and rounding them to the nearest integer, respectively.

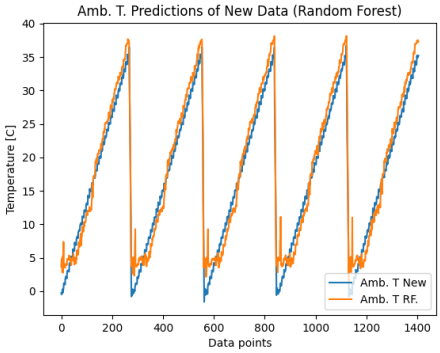

Fig. 14 Ambient Temperature Predictions, CPU=30% and Fan=70%, deletion of MV and Non-integer

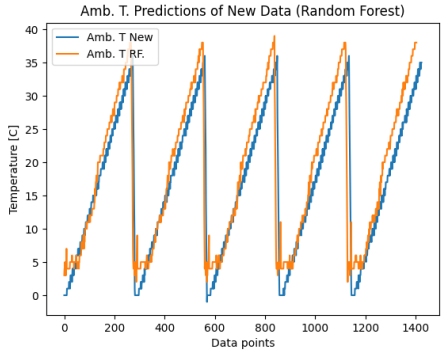

Fig. 15 Ambient Temperature Predictions, CPU=30% and Fan=70%, deletion of MV and Integer

Note that the increased number of "triangles" in the Figures only indicates consecutive test execution at the same temperature. Figure 13, 14, and 15 clearly show a case of overfitting. Possible reasons for overfitting could be:

- Insufficient training data: When the training dataset is small, the model may learn the noise or specific patterns present in the limited data. Increasing the amount of training data can help alleviate this issue.
- Feature overfitting: When the model has access to irrelevant or noisy features with no predictive power for the target variable, it may overfit by learning patterns specific to the training data. Feature selection or dimensionality reduction techniques can help address this issue.
- Complex model architecture: Models with high complexity, such as those with a large number of parameters, have a higher tendency to overfit. Simplifying the model architecture, reducing the number of parameters, or using regularization techniques can mitigate overfitting.

Table 3 shows how the value of MAE in the case of prediction based on an insufficient training dataset is higher than that obtained with an adequate number of variables in the training dataset (compare with Table 2) for both test and unseen data. An MAE greater than two indicates that, on average, the model's predictions deviate from the actual temperature by more than two degrees Celsius. This level is unacceptable; the goal is to keep the error below one degree Celsius. An R^2 of 0.94 indicates that the model is still explaining 94% of the variance in the data, which is still relatively high, but not as high as the previous value of 0.98. We calculate the updated data and prediction handling metrics, which we reported as "New" and "Int.".

Table 3 MAE and R^2 values for different models when predicting baseband ambient temperature at CPU=30% and Fan=70%

Metric	**RF Prev.**	**RF New**	**RF Int.**
Test MAE	0.897	0.893	1.032
Test R^2	0.985	0.982	0.980
Unseen MAE	2.144	1.899	2.053
Unseen R^2	0.947	0.963	0.956

4 Conclusion and Future Work

Our research used Random Forest and XGBoost Regressors to predict a baseband board's ambient temperature accurately. We exceeded previous approaches in precision, marking a significant advancement in the field. We used hyperparameters to optimize tree-based models, which are known for their suitability in regression activities, and cross-validation to evaluate the performance of the Random Forest regressor. The XGBoost Regressor and cross-validation of the Random Forest Regressor didn't show significant improvement in model performance with further data handling optimization. However, our meticulous approach to data handling optimization instils confidence in the robustness of our research, and we focused on the Random Forest Regressor only. The well-trained Random Forest model exhibited impressive accuracy, achieving a mean absolute error (MAE) of 0.654 or less and R-squared values nearing 0.987 on previously unseen data. Importantly, the model's robustness was confirmed by its improved MAE and R-squared values, signifying high confidence levels. Furthermore, permutation and feature importance analyses highlighted the key drivers of the model's performance, revealing that temperature sensors significantly influenced the predictions. Conversely, it was determined that power and voltage readings could be safely excluded from the model's attributes as they do not significantly impact temperature prediction.

Further investigation into data handling and presentation of the ambient temperature predictions was done. We could conclude from the analysis that deletion of missing values did not greatly affect the accuracy of the predictions. Looking at how the metrics changed with what data was rounded up, we can see that the MAE increased both when only input data was rounded up and when input data along the resulting predictions were converted to nearest integer but the R^2-value remained approximately the same. Thus rounding up to the nearest integer is best to perform only on the resulting predictions and not on the input data which is used for training the model, as this can result in unforseen error propagation, which will be done for better presentation of the temperatures. Any sort of rounding up in the case of insufficient data did not help the case, the MAE grew and the R^2-value increased.

Finally, predicting the ambient temperature is the first step to putting into practice those thermal throttling and preventive maintenance policies that we have indicated as the primary objective of our research (compare with Section 1.3). Pursuing the research's goals requires future study in two different but parallel domains:

- Use the ambient temperature prediction along with system resources (computer, networking, and memory) to obtain a hardware fault prediction.
- Use the prediction of ambient temperature as a critical variable in the runtime product's life cycle evaluation as a function of the environmental parameters.

Acknowledgements. The work presented in this paper is sponsored by Ericsson AB, Mälardalen University and the Swedish Knowledge Foundation (KKS), via the industrial PhD School ARRAY.

References

[1] Nestor, M., Loredana, F., Raymond, P., Vanessa, P., Anka, R., Erik, B., John, E., Katrina, L., Terah, L., James, M., Carlos, N.J., Yoav, S., Russel, W., Jack, C.: Artificial intelligence index report 2024. Technical report, Standford University, Standford, CA (April 2024)

[2] 3GPP: Ts 22 261 - v19.1.0 - 3rd generation partnership project; technical specification group services and system aspects; service requirements for the 5g system; stage 1 (release 19) (2022)

[3] Chih-Lin, I., Kukliński, S., Chen, T., Ladid, L.L.: A perspective of o-ran integration with mec, son, and network slicing in the 5g era. IEEE Network **34**, 3–4 (2020) https://doi.org/10.1109/MNET.2020.9277891

[4] Al-Dulaimi, A., Wang, X., Chih-Lin, I.: 5G Networks: Fundamental Requirements, Enabling Technologies, and Operations Management, (2018). https://doi.org/10.1002/9781119333142

[5] Vitucci, C.: The Role of Fault Management in the Embedded System Design. Mälardalen University Press Licentiate Theses, Västerås, Sweden (2024)

[6] Vitucci, C., Sundmark, D., Jägemar, M., Danielsson, J., Larsson, A., Nolte, T.: Fault management framework and multilayer recovery methodology for resilient system. Proceeding IEEE 6th International Conference on System Reliability and Safety (ICSRS), 32–39 (2022)

[7] Das, A., Mueller, F., Rountree, B.: Aarohi: Making real-time node failure prediction feasible. 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS), 1092–1101 (2020) https://doi.org/10.1109/IPDPS47924.2020.00115

[8] Chigurupati, A., Thibaux, R., Lassar, N.: Predicting hardware failure using machine learning. 2016 Annual Reliability and Maintainability Symposium (RAMS), 1–6 (2016) https://doi.org/10.1109/RAMS.2016.7448033

[9] Das, M.K., Rangarajan, K.: Performance monitoring and failure prediction of industrial equipments using artificial intelligence and machine learning methods: A survey. Proceedings of the 4th International Conference on Computing Methodologies and Communication, ICCMC 2020,

595–602 (2020) https://doi.org/10.1109/ICCMC48092.2020.ICCMC-0000111

[10] Phuyal, S., Bista, D., Bista, R.: Challenges, opportunities and future directions of smart manufacturing: A state of art review. Sustainable Futures **2**, 100023 (2020) https://doi.org/10.1016/J.SFTR.2020.100023

[11] Cotta, J., Breque, M., Nul, L.D., Petridis, A.: Industry 5.0 towards a sustainable, human-centric and resilient european industry. European Commission Research and Innovation (R&I) Series Policy Brief (2021) https://doi.org/10.2777/308407 . Accessed 2022-10-24

[12] Camps-Mur, D., Gavras, A., Ghoraishi, M., Hrasnica, H., Kaloxylos, A., Anastasopoulos, M., Tzanakaki, A., Srinivasan, G., Antevski, K., Baranda, J., Schepper, K., Casetti, C., Chiasserini, C., Garcia-Saavedra, A., Guimares, C., Kondepu, K., Li, X., Magoula, L., Malinverno, M., Cogalan, T.: Ai and ml – enablers for beyond 5g networks (2021) https://doi.org/10.5281/zenodo.4299895

[13] Spory, E.M.: Increased high-temperature ic packaging reliability using die extraction and additive manufacturing assembly. (2016). https://doi.org/10.4071/2016-hitec-18

[14] Escudero-Mancebo, D., Fernández-Villalobos, N., Martín-Llorente, Martínez-Monés, A.: Research methods in engineering design: a synthesis of recent studies using a systematic literature review. Research in Engineering Design (2023) https://doi.org/10.1007/s00163-022-00406-y

[15] Rahman, S., Olausson, M., Vitucci, C., Avgouleas, I.: Ambient temperature prediction for embedded systems using machine learning. In: Kofroň, J., Margaria, T., Seceleanu, C. (eds.) Engineering of Computer-Based Systems, pp. 12–25. Springer, Cham (2024)

[16] Yang, X., Sang, Q., Wang, C., Yu, M., Zhao, Y.: Development and challenges of reliability modeling from transistors to circuits. IEEE Journal of the Electron Devices Society (2023) https://doi.org/10.1109/JEDS.2023.3253081

[17] O'connor, P.D.T.: Arrhenius and electronics reliability. Quality and Reliability Engineering International **5** (1989) https://doi.org/10.1002/qre.4680050402

[18] Wang, N., Li, J.Y.: Efficient multi-channel thermal monitoring and temperature prediction based on improved linear regression. IEEE Transactions on Instrumentation and Measurement **71** (2022) https://doi.org/10.1109/TIM.2021.3139659

[19] Wang, N., Zhang, J.N., Liu, Z.Y., Ding, C., Sui, G.R., Jia, H.Z., Gao, X.M.: An enhanced thermoelectric collaborative cooling system with thermoelectric generator serving as a supplementary power source. IEEE Transactions on Electron Devices **68** (2021) https://doi.org/10.1109/TED.2021.3059183

[20] Lyu, N., Jin, Y., Xiong, R., Miao, S., Gao, J.: Real-time overcharge warning and early thermal runaway prediction of li-ion battery by online impedance measurement. IEEE Transactions on Industrial Electronics (2021) https://doi.org/10.1109/TIE.2021.3062267

[21] Ozceylan, B., Haverkort, B.R., Graaf, M.D., Gerards, M.E.T.: Improving temperature prediction accuracy using kalman and particle filtering methods. (2020). https://doi.org/10.1109/THERMINIC49743.2020.9420535

[22] Prisacaru, A., Gromala, P.J., Han, B., Zhang, G.Q.: Degradation estimation and prediction of electronic packages using data-driven approach. IEEE Transactions on Industrial Electronics **69** (2022) https://doi.org/10.1109/TIE.2021.3068681

[23] Ilager, S., Ramamohanarao, K., Buyya, R.: Thermal prediction for efficient energy management of clouds using machine learning. IEEE Transactions on Parallel and Distributed Systems **32** (2021) https://doi.org/10.1109/TPDS.2020.3040800

[24] Nisce, I., Jiang, X., Vishnu, S.P.: Machine learning based thermal prediction for energy-efficient cloud computing. (2023). https://

doi.org/10.1109/ccnc51644.2023.10060079

[25] Yao, X., Omori, M., Nishi, H.: Load balancing method using server temperature prediction considering multiple internal heat sources in data centers. (2021). https://doi.org/10.1109/ICM46511.2021.9385604

[26] Durgam, S., Bhosale, A., Bhosale, V., Deshpande, R., Sutar, P., Kamble, S.: Ensemble learning for predicting temperature of heat sources for minimizing electronic failures. (2021). https://doi.org/10.1109/ICNTE51185.2021.9487663

[27] Peng, Y.H., Lee, C.M., Tung, K.Y., Chen, R.: Rack inlet temperature prediction based on deep learning. (2022). https://doi.org/10.1109/ICMT56556.2022.9997747

[28] Zhang, K., Ogrenci-Memik, S., Memik, G., Yoshii, K., Sankaran, R., Beckman, P.: Minimizing thermal variation across system components. (2015). https://doi.org/10.1109/IPDPS.2015.37

[29] Cheng, T., Du, H., Li, L., Fu, Y.: Lstm-based temperature prediction and hotspot tracking for thermal-aware 3d noc system. 2021 18th International SoC Design Conference (ISOCC), 286–287 (2021)

[30] Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016). https://doi.org/10.1145/2939672.2939785

[31] Breiman, L.: Random forests. Machine Learning **45**(1), 5–32 (2001) https://doi.org/10.1023/A:1010933404324

[32] Bates, S., Hastie, T., Tibshirani, R.: Cross-validation: What does it estimate and how well does it do it? Journal of the American Statistical Association (2023) https://doi.org/10.1080/01621459.2023.2197686

[33] Pereira, R., Couto, M., Ribeiro, F., Rua, R., Cunha, J., Fernandes, J.P., Saraiva, J.: Ranking programming languages by energy efficiency. Science of Computer Programming **205** (2021) https://doi.org/10.1016/j.scico.2021.102609