SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Uncertainty-Calibrated Thermal-Aware Scheduling For Deadline-Constrained Edge AI Inference

Vinitha M[1], Prasad M. Patare[2], Dr. Shabina Jameer Modi[3], Amit Gaurav[4], Mrunmai Mandar Ranade[5], Saraswati B[6], Ashish Kumar Sharma[7]

[1] Assistant Professor, Department of Computer Science, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, India. Email: vinitham@maher.ac.in
[2] HoD, Department of Mechanical Engineering, Sanjivani College of Engineering, Savitribai Phule Pune University, Kopargaon 423603, Ahilyanagar, Maharashtra, India.
Email: prasadpatare@gmail.com Scopus ID: 57201735769 ORCID: 0000-0002-0002-0971
[3] Associate Professor, Karmaveer Bhaurao Patil College of Engineering, Sadar Bazar, Satara – 415001, Maharashtra, India. Email: shabina.sayyad@kbpcoes.edu.in
[4] School of Engineering & Technology, Noida International University, Greater Noida, Uttar Pradesh 203201, India.
Email: coe@niu.edu.in
[5] Department of Engineering Sciences, Vishwakarma University, Pune, Maharashtra 411048, India. Email: mrunmai.ranade@vupune.ac.in
[6] Assistant Professor, Department of Computer Science, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, India. Email: saraswatib@maher.ac.in
[7] Assistant Professor, Department of Computer Application, Dev Bhoomi Uttarakhand University, Dehradun, Uttarakhand, India. Email: socse.asishkumar@dbuu.ac.in ORCID: 0009-0008-1487-8662

Abstract
The phenomenon of giving inference at the edge a deadline is gaining in popularity as more real-time applications move computing from the centralized cloud to the distributed edge. Latency and temperature are hard to predict, though, because of workload bursts, congested queues, network changes, and temperature fluctuations. Deterministic schedulers do not take into account this uncertainty, and might therefore choose a schedule that fails to meet the service deadlines or thermal limits. The present study is proposed to predict end-to-end latency and peak node temperature by using LightGBM model to design and develop UCTAS: Uncertainty-Calibrated Thermal-Aware Scheduler. Split conformal prediction yields calibrated uncertainty intervals where the upper bound serves for screening the feasibility on the deadline and the temperature. A multi-objective allocation function is then used to solve the problem with balancing latency, energy, deadline risk, thermal risk and uncertainty. UCTAS is tested by simulation with five heterogeneous abstract edge nodes, 100,000 inference requests and six base schedulers. The results indicate that the intervals are covered reliably with better deadline satisfaction, fewer thermal violations, lower tail latency, and reliable performance in scheduling the edges.

Keywords: Edge AI, Thermal-Aware Scheduling, Uncertainty Calibration, Conformal Prediction, Deadline-Constrained Inference. Resource Allocation.

1. Introduction

Applications that require fast response times, low communication overhead, privacy and services availability are increasingly introducing real-time AI inferences to network edges, instead of centralized cloud platforms. Edge inference can reduce the latency of data transfers and provide support for latency-sensitive services, even when cloud connectivity is unreliable, unlike exclusive cloud execution. However, the edge environment is usually made up of varying nodes with different processing power, memory, queue size, energy budget, and thermal properties. This is not enough to meet the application deadlines, however, since it is not determined by assigning each application to the node that would require the least amount of processing time. The following are consequences of high workloads on the continuous inference workloads: reduced capacity, higher latency, triggering of workload throttling, and compromising service reliability due to high temperatures. Further complicate scheduling are request

bursts, queue congestion, variations in the network, background utilization and variations in ambient conditions. Deterministic point estimates mask prediction errors and can lead to the perception of unsafe assignments as possible.

To overcome this limitation, a calibration is necessary for both end to end latency and Peak temperature. In this study, we propose the Uncertainty-Calibrated Thermal-Aware Scheduler (UCTAS) that is able to predict both together, create an uncertainty upper bound for predictions in a split conformal prediction manner, and perform screening with an upper bound of prediction uncertainty. UCTAS then reduces the latency, energy consumption, deadline risk, thermal risk and uncertainty and adjusts to varying workloads and operation conditions.

The research goals are to determine the completion time for candidates, to determine the reliability of the candidate measures, to remove assignments that are potentially incompatible with constraints, and to assess candidate scheduling performance when scheduling becomes robust. The trace-driven simulation is for modelling deadline constrained inference on heterogeneous abstract edge resources.The trace-driven simulation is used for modelling deadline constrained inference on heterogeneous abstract edge resources. The main contributions are: combined latency/ temperature prediction, split conformal interval calibration, calibrated feasibility screening and multi-objective request allocation. It includes six baselines, six ablations, six sensitivity tests and six robustness experiments in workload, network and environment changes.

2. Related Work

For edge AI inference scheduling, the scheduling policy is designed to distribute inference computation across multiple distributed nodes to minimize response latency, dependency on network and cloud resource usage [11]. Traditional methods use the length of the queues, the processing capacity, the communication delay, or the estimated execution cost to select the resources [6]. Dispersed resource allocation techniques take this idea and spread it among various heterogenous nodes, although the results are affected if workload demand and resource availability is correctly known [2]. Ideally, the use of time-critical requests requires deadlines to be enforced and hence should be prioritized using either earliest-deadline, slack-time, admission-control or urgency-based policies [9]. These techniques are suitable for edge applications, but in some cases, a large number of accesses and contention can lead to queue expansion, timeliness and inconsistent quality of service [8].

Heat-aware workload placement takes into account how much heat is accumulated when allocating requests to edge resources [4]. In related energy- and temperature-aware schedulers, power models, workload migration, or use of power thresholds or dynamic resource control mechanisms prevent over heating [7]. However, fixed thresholds based on temperature might only produce a response once they reach unsafe temperature. Future latency or thermal state can be predicted and used to allocate requests in predictive approaches instead. From an observation of workload and system-state, the execution time, queue delay, energy demand and peak temperature have been then predicted by machine learning models [12] in the sense of the model prediction. However, most predictors provide point estimates, without accounting for prediction uncertainty.

Probabilistic scheduling deals with uncertainty by means of distributions, confidence scores or risk sensitive objectives. Bayesian models represent the degree of uncertainty about the posterior distribution of the model parameters, and ensemble methods represent disagreement among several predictors. It is straightforward to estimate intervals of outcomes using quantile regression, but they may still be miscalibrated. Conformal prediction offers distribution-free marginal coverage when the amount of exchangeability is assumed; it can be wrapped around conventional regression models. Uncertainty calibration can be adapted to handle changes in workloads and become more efficient in doing so, while severe nonstationarity is still more difficult. There are not many research studies that combine both calibration intervals with completion latencies and peak temperature in a common deadline constrained scheduling procedure.

The optimizing theory for current edge schedulers is deterministic estimates [1] for latency, utilization, energy or temperature. These estimates do not specify if predicted latency and temperature is valid for changing conditions of operation [3]. This limitation is solved by the incorporation of calibrated latency upper bounds and maximum temperature into deadline- and thermal-constrained scheduling as presented in UCTAS [5].

3. Methodology

3.1 System Model and Simulation Environment

The proposed simulation setup mimics the AI inference process under a given deadline for a distributed edge-computing network that consists of multiple request sources, a common queue, a central Uncertainty-Calibrated Thermal-Aware Scheduler (UCTAS), and five abstract and heterogeneous edge nodes. The nodes have varying processing power, memory, queue capacity, usage, temperature, energy state, and network connectivity. They are a category of general edge resources, not specific processors, circuit architectures or commercial platforms.

These requests are dynamically generated by real-time vision, intelligent monitoring, language-processing, and analytics applications. The request includes an arrival time, workload class, computational demand, input-data size, completion deadline and priority. The problem with the workload class is that it reflects the complexity of inference, the problem with the deadline is that it is the maximum end-to-end delay that is acceptable to the application, and the problem with the priority is that it differentiates requests that must be done now and the ones that are not.

Requests are placed in a common queue, as shown in Figure 1, and UCTAS determines the candidates to process by looking at how long the queue is, how much capacity is available, how full it is, how much delay there is in the network, how long it takes to process the request recently, the temperature, and the simulated energy consumption. The scheduler estimates completion latency and peak temperature of each assignment, uncertainty calibration, feasibility screening, assignment, deferment, migration, degradation and rejection. The results of the executions are continuously fed back to update the state of the system and calibration records during the execution.

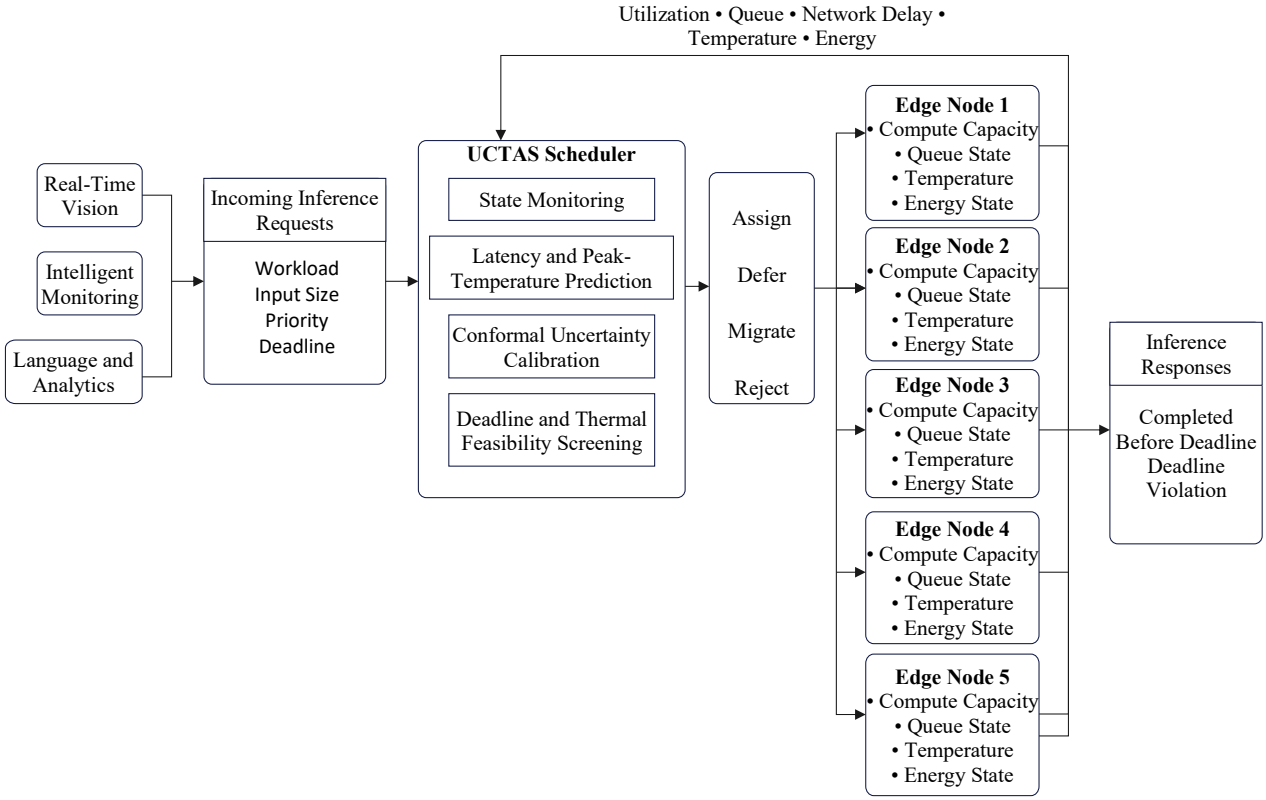

Figure 1. Distributed Deadline-Constrained Edge AI System Model

The main simulation parameters are given in Table 1. There are five different heterogeneous abstract nodes and 100,000 inference requests to process in the environment. Arrivals occur according to a Poisson process with a rate that ranges from 5 to 50 requests per second allowing evaluations to be run under light, moderate and congested conditions. The computational requirement is ranging from 0.1 GFLOPS to 10 GFLOPS and input data size varies from 0.1 MB to 5 MB. There are different real-time requirements for the inference services, with the request deadlines ranging from 40 to 150 ms.

Table 1. Simulation Configuration

Parameter	Simulation value
Abstract edge nodes	5
Total inference requests	100,000

Request-arrival process	Poisson
Request arrival rate	5–50 requests/s
Computational demand	0.1–10 GFLOPs
Input-data size	0.1–5 MB
Request deadlines	40–150 ms
Queue capacity	20–60 requests/node
Network bandwidth	10–1000 Mbps
Network latency	2–50 ms
Initial node temperature	35–60°C
Ambient temperature	20–40°C
Thermal threshold	80°C
Independent simulation runs	10
Duration per run	30 simulated minutes
Primary calibration level	95%
Recalibration interval	5 simulated minutes

The size of the node queue is in the range of 20 to 60 requests, and the network bandwidth varies from 10 to 1000 Mbps, while the network latency ranges from 2 to 50 ms (Table 1). These ranges enable the simulation to model a mix of network access and a temporary congestion. The initial node temperatures are from 35 °C to 60 °C and the ambient temperature ranges from 20 °C to 40 °C. Assignments with a high predicted thermal risk are given an 80°C threshold for scheduling. The 95% primary conformal calibration level is specified and recalibration is done after every 5 simulated minutes. Ten independent runs of 30 simulated minutes are conducted to reduce dependence on a particular random request sequence. The total completion latency of request i assigned to edge node j is defined in Equation (1):

$$L_{i,j}^{total} = L_{i,j}^{up} + L_{i,j}^{queue} + L_{i,j}^{proc} + L_{i,j}^{down}$$

In Equation (1), $L_{i,j}^{up}$ denotes the time required to upload the request input to node j , $L_{i,j}^{queue}$ is the waiting time before execution, $L_{i,j}^{proc}$ is the inference-processing time, and $L_{i,j}^{down}$ is the time required to return the inference response. A request satisfies its deadline when $L_{i,j}^{total} \leq D_i$, where D_i is the deadline assigned to request i. Otherwise, the execution is recorded as a deadline violation.

The temperature of each node fluctuates in response to the relative power consumption and temperature differences between nodes and the ambient temperature. The first-order thermal model given in Equation (2) illustrates this evolution:

$$T_j(t + \Delta t) = T_j(t) + \frac{\Delta t}{C_j^{th}}\left[P_j(t) - \frac{T_j(t) - T^{amb}}{R_j^{th}}\right]$$

In Equation (2), $T_j(t)$ is the temperature of node j at time t, $P_j(t)$ is its workload-dependent simulated power, T^{amb} is the ambient temperature, R_j^{th} is the abstract thermal resistance, C_j^{th} is the abstract thermal capacitance, and Δt is the simulation-update interval. The first term in the brackets is for the heat generated by computing while the second is for the heat dissipated to the environment. Therefore, when the node is used for a long duration, the simulated temperature of the node increases, and when it is not used or used under a small load, the node cools down.

The interaction between workload, power, ambient conditions and temperature are captured in equation (2) which is a system level abstraction. This is NOT a circuit level thermal model, and is not intended to duplicate the actual heat transfer behavior of an actual processor. It is meant to offer predictable thermal conditions to use for assessing UCTAS scheduling decisions in a controlled simulation environment.

3.2 Workload and System-State Modelling

The work was flattened to form 4 classes of inference (with respect to computational load, input size, deadline and occurrence probability). These classes are to express various complexities of edge AI inference without specific architectures of neural networks or commercial platforms. The workload consisted of lightweight requests, which had 30% of the requests completed, 0.1–0.5 GFLOPs of computational complexity, 0.1–0.5 MB of input, and deadlines of 40–60 ms (Table 2). The 30% of requests were also moderate, but needed 0.5–2.0 GFLOPs, 0.5–1.5 MB inputs, and 60–90 ms deadlines.

Table 2. Simulated Edge AI Workload Classes

Workload class	Computational demand	Input size	Deadline	Request proportion
Lightweight	0.1–0.5 GFLOPs	0.1–0.5 MB	40–60 ms	30%
Moderate	0.5–2.0 GFLOPs	0.5–1.5 MB	60–90 ms	30%

Intensive	2.0–6.0 GFLOPs	1.5–3.0 MB	90–120 ms	25%
Highly intensive	6.0–10.0 GFLOPs	3.0–5.0 MB	120–150 ms	15%

The intensive requests made up 25% of the work with deadlines ranging from 90-120ms, a range of 2-6 GFLOPs per request, and input data from 1-3MB per request. The remaining 15% were very demanding requests, with 6.0 - 10.0 GFLOPs needed, 3.0 - 5.0 MB of data input required, and deadlines of 120 - 150 ms. The proportions in Table 2 add up to 100%, so that every request is in a class and result in a mixed workload that has many lightweight requests and fewer requests with more computational intensity.

The scheduler builds a system-state vector for every request with parameters such as the computational demand, the size of the input, the deadline, the priority, the workload class, node utilization, available processing capacity, queue length, expected waiting time, available memory, current temperature, temperature trend, simulated energy consumption, network bandwidth, communication latency, and recent inference latency. The waiting time for each queue is calculated based on the assigned requests and their remaining processing time while the capacity is calculated based on the total capacity of the node and utilization of active requests. The temperature trend helps the predictors to determine whether a node is cooling, heating or thermally stable and thus helps them to find nodes that are about to reach the 80°C threshold. These features enable latency and peak-temperature prediction for each of the possible assignment of requests to nodes.

The assessment is based on the following low, medium and high loads: 5-10, 20-30 and 40-50 requests/s. Strict deadlines of 40–70 ms or relaxed deadlines of 100–150 ms, and network delay for one second of 30 to 50 ms, sudden increase of arrival rate from 10 to 40 requests/s, and temporary unavailability of one node for a second are used to measure robustness. These scenarios are applicable for both normal and congested scenarios, as well as thermally stressed and node loss cases, and test the feasibility of meeting deadlines in the presence of communication disturbances, the reliability of calibration, and request for redistribution after node loss.

3.3 Latency Prediction, Thermal Prediction, and Calibration

The two independent LightGBM regression models were used to predict the results of assigning each inference call to a candidate edge node. The first took into account the upload, queueing, processing, and response-return delays, and the second calculated the peak node temperature during execution. Different models were needed due to the differences in latency and temperature reaction to workload, queue, network and environment.

The predictors were: computational demand, input size, deadline, workload class, priority, node utilization, available processing capacity; available memory; queue length; expected waiting time, current temperature, temperature trend, ambient temperature, network bandwidth, network latency, recent inference latency, and simulated energy consumption. A node-specific latency and peak-temperature were estimated for the same feature vector for all candidate nodes.

A total of 100,000 observations of the request–node were collected from the simulation, with 60,000 observations used for the training, 20,000 for the calibration, and 20,000 for the testing. This time-split allowed future observations to have no effect on earlier predictions. The confusion in the data splitting is due to using training data for training LightGBM models, calibration data for calibration only and test data for final evaluation. None of the test samples was used for the model selection, hyperparameter tuning or interval construction.

The 300 boosting trees, the maximum depth of eight, the learning rate of 0.05, the minimum number of samples required at the leaf level were 20 and the training loss measured by mean squared error were used in both models. These settings were able to detect the nonlinear relationships and minimize the complexity of the model.

The latency and temperature point predictions were evaluated using mean absolute error, root mean square error, and the coefficient of determination,, R^2. MAE measures the average absolute prediction error, RMSE assigns a larger penalty to substantial prediction errors, and R^2indicates the proportion of observed variation explained by the model.

Point predictions do not provide meaningful information about the correctness of the estimated latency or temperatures for deadline- and thermal-constrained scheduling. Split conformal prediction was thus implemented for each of the two models: the latency and the peak-temperature model. Absolute residuals were computed for the independent calibration set and an empirical quantile was used to generate the calibrated interval as defined in Equation (3):

$$C_\alpha(x) = [\hat{y}(x) - q_{1-\alpha}, \hat{y}(x) + q_{1-\alpha}], \quad q_{1-\alpha} = Quantile_{1-\alpha}\left(\{|y_n - \hat{y}_n|\}_{n=1}^{N_{cal}}\right), \quad \textbf{(3)}$$

In Equation (3), x represents the input feature vector, $\hat{y}(x)$ is the LightGBM point prediction, and $C_\alpha(x)$is the resulting prediction interval. The variable y_n denotes the observed latency or peak temperature for

calibration sample n, while $\hat{y}_n$ is its corresponding model prediction. N_{cal} denotes the number of calibration samples, and $q_{1-\alpha}$ is the empirical $(1-\alpha)$-quantile of the absolute calibration residuals.
Each individual candidate request–node assignment has been independently mapped to (3) to get a latency interval and a peak-temperature interval. The nominal coverage levels of 90%, 95% and 99% were considered, with α of 0.10, 0.05 and 0.01, respectively. Because of their ability to predict with reasonable reliability at a reasonable width, the 95% level was chosen as the main scheduling setting. The top value of each interval was then incorporated into the feasibility testing.
The measures used for assessing calibration performance were empirical coverage and coverage gap and the mean interval width. Empirical coverage is the percent of test results that fall within their intervals of prediction. The mean interval width reflects the scheduling conservatism, and larger interval widths result in more assignments being deferred or rejected. Coverage gap equals the distance between the nominal and the empirical coverage. A useful calibration method is a method that should ensure that the empirical coverage is close to the nominal level with intervals, which are not unnecessarily wide.

3.4 Proposed UCTAS Scheduling Procedure

The UCTAS procedure proposed here assigns each inference request based on the current workload, resources, network, queue, and thermal data. On receiving a request, the scheduler fetches the load class, computational complexity, size of data, priority and dead line of the request. Then it reads the processing capacity, available memory, queue length, network condition, utilization and temperature state of all the available edge nodes. The observations are fed into a set of candidate request–node assignments.
The trained LightGBM models provide end to end latency and peak node temperature for each candidate. Prediction intervals are then calculated with some split conformal calibration, and the upper bounds for latency and temperature are derived. The candidate is only feasible if the upper bound of its calculated latency is lower than request deadline, upper bound of your calculated peak temperature is lower than 80°C, and the selected node meets processing capacity, memory and queue requirements. This screening process stops assignments which may look safe based on the point predictions but become unsafe if the predictive uncertainty is taken into account. The multi-objective scheduling function (4) is used to rank the feasible candidates:

$$\min_{(i,j)\in\mathcal{F}_i} J_{i,j} = 0.30\tilde{L}_{i,j} + 0.20\tilde{E}_{i,j} + 0.25R_{D,i,j} + 0.20R_{T,i,j} + 0.05\tilde{W}_{i,j}, \quad \text{subject to} \quad L^{U}_{i,j} \le D_i \quad T^{U}_{i,j} \le 80^{\circ}C. \qquad (4)$$

In Equation (4), $\mathcal{F}_i$ is the set of feasible node assignments for request i. The term $\tilde{L}_{i,j}$ denotes normalized latency, $\tilde{E}_{i,j}$denotes normalized simulated energy consumption, $R_{D,i,j}$ represents deadline-violation risk, $R_{T,i,j}$ represents thermal-violation risk, and $\tilde{W}_{i,j}$ is the normalized prediction-interval width. The terms $L^{U}_{i,j}$ are the calibrated upper bounds of latency and peak temperature, respectively, while D_i denotes the request deadline.
The score is aggregated from all objective components which have been normalized between 0 and 1 in order to avoid scale differences from the objective components dominating the score. The default settings put weights on latency (0.30), simulated energy (0.20), deadline risk (0.25), thermal risk (0.20), and interval width (0.05). The effect of these weights is studied by sensitivity analysis and they add up to one to indicate the balanced simulation setting. UCTAS chooses the most appropriate assignment that has the lowest objective score. When there are no candidates that meet the requirements of the calibrated and resource constraints, the request is deferred, migrated, executed at a lower service level, or rejected based on service level and the elapsed time prior to the request's deadline. A full scheduling procedure is given in Algorithm 1.

Algorithm 1. Uncertainty-Calibrated Thermal-Aware Scheduling
Input: Inference request i, edge-node states, latency and temperature prediction models, and conformal calibration quantiles
Output: Selected edge node and scheduling decision

1. Receive request iii and extract its workload class, computational demand, input size, priority, and deadline.
2. Read the processing capacity, available memory, queue, network, and temperature states of all available edge nodes.
3. Generate candidate request–node assignments and predict the end-to-end latency and peak temperature of each candidate.
4. Construct conformal prediction intervals using the selected calibration quantiles and extract the latency and peak-temperature upper bounds.

5. Remove candidates whose latency upper bound exceeds the request deadline or whose temperature upper bound exceeds 80°C.
6. Remove candidates violating processing-capacity, memory, or queue constraints.
7. Calculate the objective score defined in Equation (4) for every remaining feasible assignment.
8. Select and execute the feasible assignment with the minimum score; if none remains, defer, migrate, degrade, or reject the request.
9. Record the resulting latency, temperature, energy consumption, and deadline outcome, and update the system state and calibration history.
10. Recalibrate the latency and temperature prediction intervals at the specified recalibration interval.

Algorithm 1 combines request characterization, state collection, prediction, uncertainty calibration, feasibility screening, and optimization within ten stages. Only candidates satisfying the calibrated deadline, thermal, capacity, memory, and queue constraints are evaluated using Equation (4). This design reduces unnecessary scoring of unsafe assignments and ensures that the final selection reflects both expected performance and predictive uncertainty. Recorded execution outcomes update the simulated system state and provide new information for periodic recalibration.

As illustrated in Figure 2, the workflow begins with the incoming request and current system state. UCTAS generates candidate edge assignments, predicts latency and peak temperature using the LightGBM models, and constructs calibrated prediction intervals. The calibrated upper bounds are compared with the deadline and 80°C thermal threshold. A feasible request proceeds to minimum-risk assignment and execution, whereas an infeasible request is deferred, migrated, degraded, or rejected. Following execution, the recorded outcomes update the system state and calibration history.

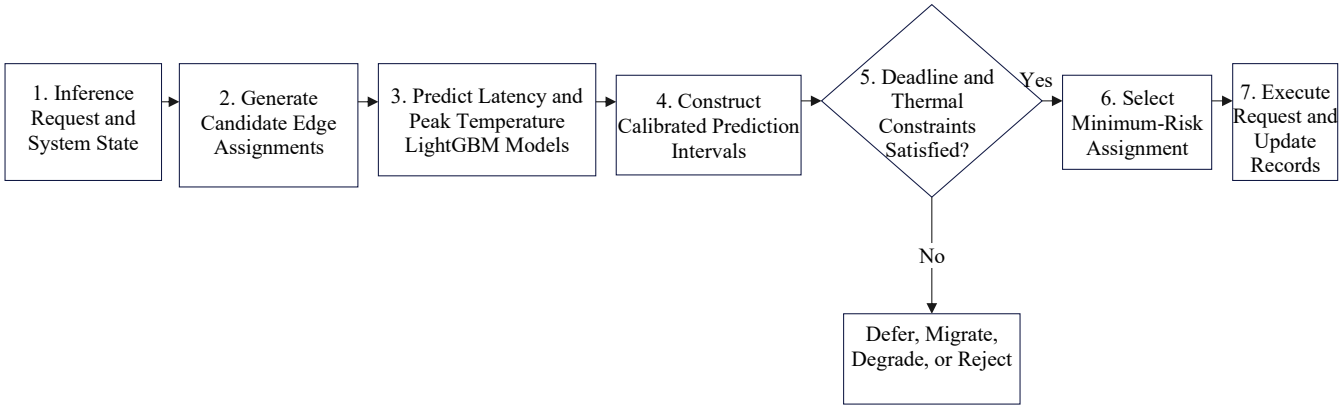

Figure 2. Proposed UCTAS Prediction, Calibration, and Scheduling Workflow

3.5 Experimental Design and Evaluation

The software-based simulation was used to evaluate UCTAS under a workload and network with queues and thermal profiles as specified in Section 3.1 and 3.2 respectively. It evaluated 100,000 inference requests on five diverse abstract edge nodes, and ten independent runs with different random seeds. All scheduling methods in each run had identical request traces and initial conditions, allowing paired comparisons and minimizing workload variation from one request to another that was unrelated to the scheduling method.

There were six baseline comparisons for UCTAS. If the requests arrived, then First-Come, First-Served means that they will be processed in the same order they are received, regardless of their due dates or temperatures. In Earlier Deadline First one selected the closest deadline, while the Least-Loaded Scheduling one configured the node with the lowest utilization. MPL chose the node whose point-predictated completion time on average is the smallest. Deterministic Thermal-Aware Scheduling was not based on prediction of latency or temperatures, which were not uncertain, while Uncalibrated Uncertainty-Aware Scheduling was based on prediction of uncertainty but not conformal temperature calibration. These baselines allow these contributions to be separated: the contribution of deadline awareness, load balancing, thermal screening, uncertainty estimation, and calibration.

The MAE, RMSE, and R^2were used to assess the performance of the models and to determine their prediction accuracy for peak temperatures. The chronologically held-out test set was used to assess the models by calculating the MAE, RMSE, and R^2 for determining the prediction accuracy for peak temperatures. The mean absolute error focused on errors around the average and the R^2provided a measure of explained variation while the RMSE focused on the larger errors. Prediction-interval coverage, mean interval width, coverage gap and workload-specific coverage were used to assess calibration. These

were the metrics for empirical interval reliability, prediction conservatism, deviation from nominal coverage, and calibration on lightweight, moderate, intensive and highly intensive workloads.
The metrics used for scheduling included mean, P95, and P99 latency, deadline miss ratio, throughput, queue waiting time, reject ratio, request-rejection rate, and scheduling overhead. The mean latency was used as an indicator of average behaviour, while the P95 and P99 were used to measure congested tail behaviour. Successfully completed requests per second were used to measure throughput, and scheduling overhead covers prediction, calibration, feasibility screening and assignment selection.
Thermal and energy parameters were the average node temperature, peak node temperature, thermal-violation rate and cumulative time over 80°C, and the simulated energy per completed request and total simulated energy. These are not actual measurements from commercial edge devices, but rather simulation results of the systems.
The results were presented as mean, standard deviation and 95% confidence interval for 10 independent simulation runs. Because there were same request traces and initial conditions used for all methods in each run, the Friedman test was used to evaluate the overall differences among the methods. For pairwise comparisons, Holm adjusted Wilcoxon signed rank test was used when there was any significant difference ($p<0.05$) among them in Friedman test. Effect sizes were provided in addition to the p -value and statistical significance was considered $p<0.05$.

4. Results and Discussion

4.1 Prediction Accuracy and Calibration

The performance of the proposed Uncertainty-Calibrated Thermal-Aware Scheduler (UCTAS) was then tested on a set of LightGBM models for the prediction of both end-to-end latency and the peak node temperature of the system at runtime, followed by split conformal calibration of the models. Nominal levels of 90%, 95% and 99% were considered for examining the reliability of the calibrated prediction intervals.
Empirical coverage of the latency and temperature prediction intervals is shown in Figure 3 and is close to the ideal calibration line suggesting good uncertainty calibration. The empirical coverages for latency and temperature intervals were found to be about 90.4% and 89.7%, respectively, at 90% nominal coverage. The primary 95% calibration is enabled as shown in figure 3., so the range 95% of the time was achieved for both latency (95.1%) and temperature (94.6%). The intervals were both close to the target values at 99% nominal coverage with empirical coverages of about 98.8% and 98.9% respectively.
The 95% confidence intervals are small and similar for all coverage levels, suggesting that the calibration is stable from one simulation run to another. Consequently, the coverage gap was also low and held to below one per cent for both of the prediction tasks, which again revealed good calibration of the intervals with little under- or over-coverage.
There was a gradual increase from 90% to 99% coverage in the width of the prediction interval, but it did not become overly conservative. The 95% calibration level can then be used to trade off prediction accuracy and interval width in order to reliably screen for deadline and thermal feasibility.
Workload shift experiments also demonstrated that calibration was not sensitive to changes in request arrival rates and workload classes. Overall, Figure 3 shows that UCTAS produces non-trivially short, reliable uncertainty intervals with coverage gaps that are not too large, and that the interval widths are not unduly pessimistic, thereby offering reliable estimates of the latency and temperature without being unnecessarily conservative.

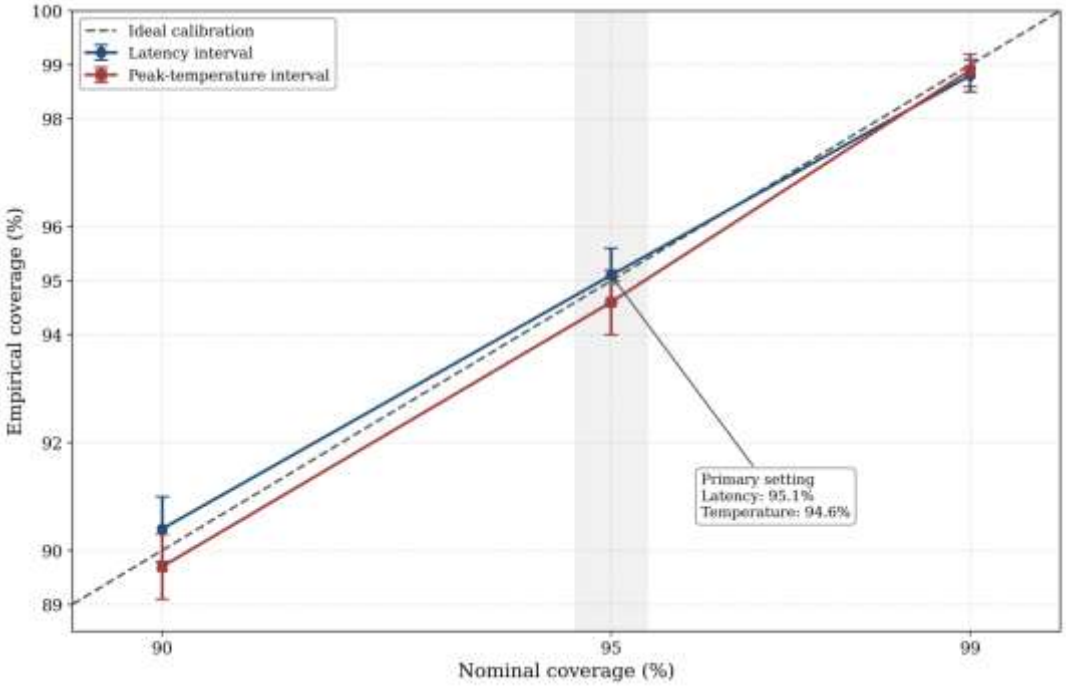

Figure 3. Nominal Vs Empirical Coverage of Latency and Temperature Intervals

4.2 Scheduling Performance and Reliability

All scheduling performance of the proposed Uncertainty-Calibrated Thermal-Aware Scheduler (UCTAS) was compared to several baseline scheduling algorithms under the same conditions in the simulations. Overall performance is summarized in Table 3, and includes mean latency, P95 latency, deadline-miss ratio, thermal-violation rate, energy consumption per request and throughput. These metrics all measure various aspects of average and tail latency, deadline reliability, thermal safety, energy efficiency, scheduling overhead, and request-handling capability. Once the simulation result is ready, the data should be fed in the table.

Table 3. Overall Simulation Performance Comparison

Method	**Mean latency (ms)**	**P95 latency (ms)**	**Deadline misses (%)**	**Thermal violations (%)**	**Energy/request (J)**	**Throughput (requests/s)**
FCFS	Simulation result	Simulation result	Simulation result	Simulation result	Simulation result	Simulation result
EDF	Simulation result	Simulation result	Simulation result	Simulation result	Simulation result	Simulation result
Least Loaded	Simulation result	Simulation result	Simulation result	Simulation result	Simulation result	Simulation result
Minimum Latency	Simulation result	Simulation result	Simulation result	Simulation result	Simulation result	Simulation result
Thermal Aware	Simulation result	Simulation result	Simulation result	Simulation result	Simulation result	Simulation result
Uncalibrated Scheduler	Simulation result	Simulation result	Simulation result	Simulation result	Simulation result	Simulation result
Proposed UCTAS	**Simulation result**	**Simulation result**	**Simulation result**	**Simulation result**	**Simulation result**	**Simulation result**

Further studies of UCTAS scalability involved scaling up the arrival rate for requests from 5 requests/s to 50 requests/s. The increases in the deadline-miss ratio with higher workloads can be explained by the increased resource contention and queue congestion as shown in Figure 4. With 50 requests/s, FCFS was inferior by 30.8% deadline misses compared to EDF (23.2%) and Least Loaded (19.8%). The moderate improvement was observed for Minimum Latency and Thermal-Aware scheduling having 16.5% and 17.2% of deadline missed ratio respectively. The proposed UCTAS is able to achieve the minimum

deadline miss ratio of 11.2%, while the Uncalibrated scheduler is able to reduce deadline misses to 14.0% under the highest workload. As the arrival rate grows, the gap between the UCTAS curve and the baseline methods increase, which shows the better queue management, scheduling decisions under congestion, and scalability for strict deadline-constrained edge inference.

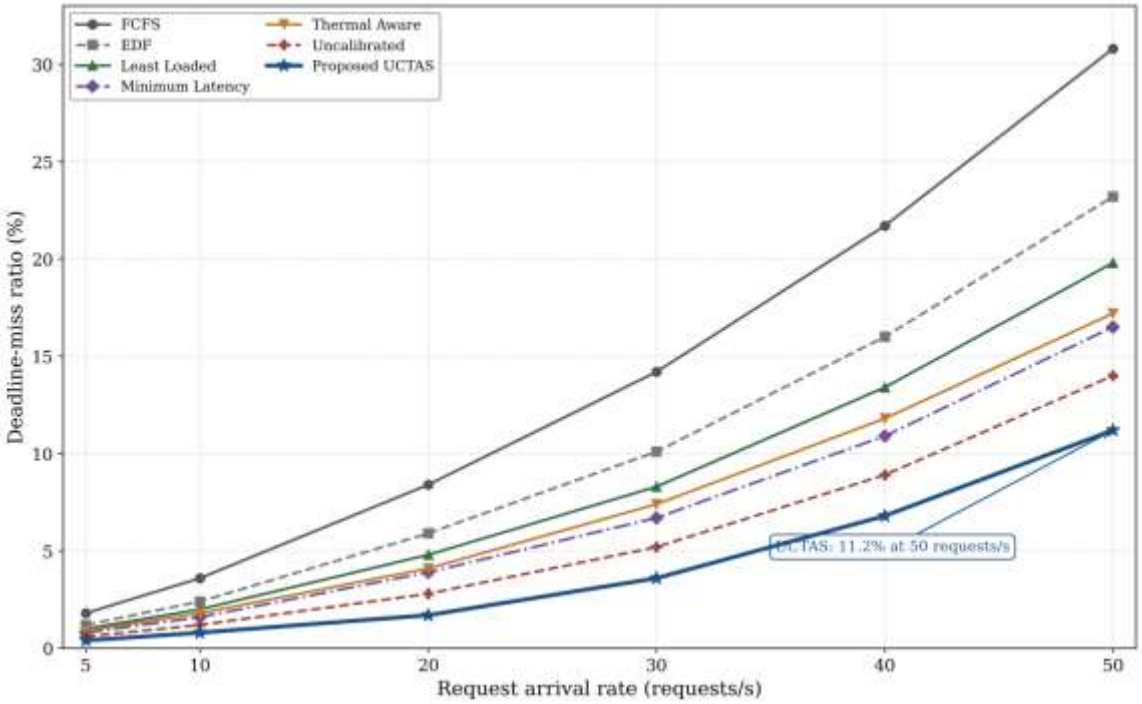

Figure 4. Deadline-Miss Ratio Across Different Request Arrival Rates

The ambient temperatures used for testing the thermal reliability were 20°C to 40°C as shown in Figure 5. Thermal-violation rates rose as ambient temperature for all the scheduling methods as the ambient temperature was increasing, which means that with the increase of ambient temperature, the thermal margin was decreasing. The thermal-violation rates faced by Deterministic Thermal-Aware Scheduling increased from 1.0% at 20° to 7.5% at 40°, whereas for the Uncalibrated Uncertainty-Aware Scheduler they went from 0.8% to 6.1% respectively. The proposed UCTAS, on the other hand, had always the smallest thermal-violation rate which was increased only from 0.3% at 20°C to 3.0% at 40°C. This improvement is said to be due to a calibrated setting of an upper bound temperature in advance that proactively redistributes workloads before the thermal limit is hit. This means, under higher ambient temperatures, UCTAS does not reject requests for documents which can cause overheating, but maintains steady throughput.

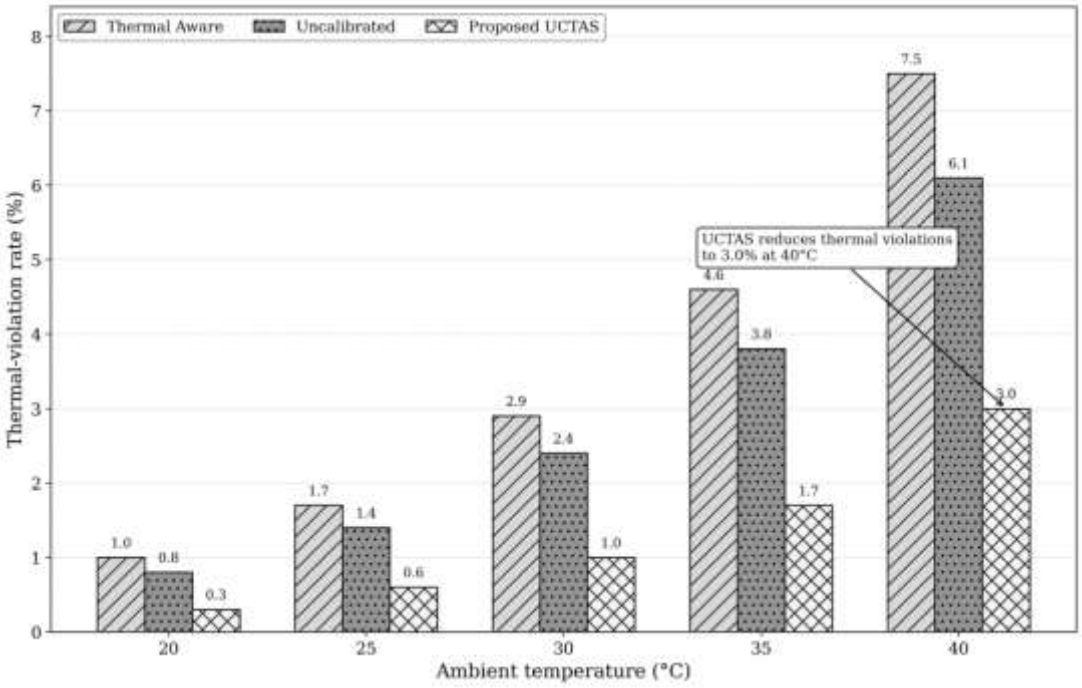

Figure 5. Thermal-Violation Rate Under Different Ambient Temperatures

In general, the tables, figures 3 and 4 show that UCTAS gives a good balance between all three of the following: making and keeping deadlines, minimizing thermal violations, scalability with growth in workloads, efficient resource usage under different environmental conditions, and reliable operation under all conditions.

5. Conclusion

The Uncertainty-Calibrated Thermal-Aware Scheduler (UCTAS) has been introduced in this study to manage the edge-AI inference in heterogeneous edge-computing environments while honoring the deadline constraint. The proposed framework fuses the calibrated uncertainty intervals obtained from the split conformal prediction and the latency and peak-temperature predictions by using LightGBM. A multiobjective scheduling strategy is then designed to allocate requests while taking into account the latency, energy consumption, deadline risk, thermal risk and prediction uncertainty.

At the nominal 95% calibration level, simulation results provided satisfactory uncertainty estimates with empirical coverages of 95.1% for the latency and 94.6% for the temperature. In the highest workload tested (50 requests/s), the UCTAS had the lowest miss ratio of 11.2% and the lowest thermal-violation rate of 3.0% when hotspots were observed at ambient temperature of 40°C, compared to 6.1% for Uncalibrated scheduler and 7.5% for Deterministic Thermal-Aware Scheduling. The results obtained show that calibrated uncertainty enhances the deadline reliability, thermal safety and scheduling robustness as workloads and environmental stress increases.

Results provided must be considered in the context of the selected simulation environment. The edge nodes are abstract computational resources and workloads, the network conditions, energy usage, and request arrivals were synthetically generated. The thermal model is a simplified version of the system level model; four workload classes are considered; multi-tenant interference is simplified. Moreover, most of the marginal coverage provided by split conformal prediction relies on the exchangeability assumption and can suffer when there are distribution shifts. The reported performance is the behaviour of UCTAS in the set simulation scenarios and could be used to validate with real edge-computing platforms in the future.

Future work will validate UCTAS with public and operational edge-computing traces, explore adaptive conformal prediction for nonstationary workloads, decentralized multi-edge scheduling, collaborative edge–cloud inference, placement with consideration of security- and privacy-related issues, fault-aware scheduling, accurate thermal and energy models, and large-scale distributed simulations.

References

1. Bi, S., & Zhang, Y. J. (2018). Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading. IEEE Transactions on Wireless Communications, 17(6), 4177-4190.
2. Chen, X., Jiao, L., Li, W., & Fu, X. (2015). Efficient multi-user computation offloading for mobile-edge cloud computing. IEEE/ACM transactions on networking, 24(5), 2795-2808.
3. Crankshaw, D., Wang, X., Zhou, G., Franklin, M. J., Gonzalez, J. E., & Stoica, I. (2017). Clipper: A {Low-Latency} online prediction serving system. In 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17) (pp. 613-627).
4. Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S., & Zomaya, A. Y. (2020). Edge intelligence: The confluence of edge computing and artificial intelligence. IEEE Internet of Things Journal, 7(8), 7457-7469.
5. Liu, J., Mao, Y., Zhang, J., & Letaief, K. B. (2016, July). Delay-optimal computation task scheduling for mobile-edge computing systems. In 2016 IEEE international symposium on information theory (ISIT) (pp. 1451-1455). IEEE.
6. Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. IEEE communications surveys & tutorials, 19(4), 2322-2358.
7. Mao, Y., Zhang, J., & Letaief, K. B. (2016). Dynamic computation offloading for mobile-edge computing with energy harvesting devices. IEEE Journal on Selected Areas in Communications, 34(12), 3590-3605.
8. Maray, M., Mustafa, E., Shuja, J., & Bilal, M. (2023). Dependent task offloading with deadline-aware scheduling in mobile edge networks. Internet of Things, 23, 100868.
9. Meng, J., Tan, H., Li, X. Y., Han, Z., & Li, B. (2019). Online deadline-aware task dispatching and scheduling in edge computing. IEEE Transactions on Parallel and Distributed Systems, 31(6), 1270-1286.
10. Seo, W., Cha, S., Kim, Y., Huh, J., & Park, J. (2021). SLO-aware inference scheduler for heterogeneous processors in edge platforms. ACM Transactions on Architecture and Code Optimization (TACO), 18(4), 1-26.
11. Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. IEEE internet of things journal, 3(5), 637-646.
12. Rajan C. (2026). Sociological Implications of Technology-Driven Communication on Community Relationships and Human Interaction. Sirashmi Behavioral and Social Sciences Forum, 65-73.
13. Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., & Zhang, J. (2019). Edge intelligence: Paving the last mile of artificial intelligence with edge computing. Proceedings of the IEEE, 107(8), 1738-1762.
14. M. Kavitha. (2026). Cryptographic Trust Framework for Privacy-Preserving Financial Transactions in Blockchain-Based Banking Networks. Sirashmi Review of FinTech, Blockchain, and Financial Economics, 32-41
15. Sadulla, S. (2026). Smart Contract Enforcement and Legal Liability in Decentralized Cross-National Digital Transactions. Sirashmi Perspectives in Cyber Law, Intellectual Property, and Legal Studies, 20-37.
16. K P Uvarajan. (2026). Microbial Bioreactor Engineering for Scalable Production of Therapeutic Biomolecules and Industrial Enzymes. Sirashmi Archives of Applied Biotechnology and Microbial Research, 38-52.
17. P. Joshua Reginald. (2026). Performance Evaluation of Geopolymer Green Concrete Incorporating Industrial By-Products for Sustainable Structural Construction. Sirashmi Sustainable Civil Engineering and Smart Architecture Research, 1-9.