Prediction of Thermal Hazards in a Real Datacenter Room Using Temporal Convolutional Networks

Prediction of Thermal Hazards in a Real Datacenter Room Using Temporal Convolutional Networks

Mohsen Seyedkazemi Ardebili*, Marcello Zanghieri*, Alessio Burrello*, Francesco Beneventi*,
Andrea Acquaviva*, Luca Benini*†, and Andrea Bartolini*
*Department of Electrical, Electronic and Information Engineering (DEI) "Guglielmo Marconi"
Università degli Studi di Bologna, Bologna, Italy
{mohsen.seyedkazemi, marcello.zanghieri2, alessio.burrello, francesco.beneventi,
andrea.acquaviva, luca.benini, a.bartolini}@unibo.it
†Integrated Systems Laboratory, ETH Zurich, Switzerland, lbenini@iis.ee.ethz.ch

***Abstract*—Datacenters play a vital role in today's society. At large, a datacenter room is a complex controlled environment composed of thousands of computing nodes, which consume kW of power. To dissipate the power, forced air/liquid flow is employed, with a cost of millions of euros per year. Reducing this cost involves using free-cooling and average case design, which can create a cooling shortage and thermal hazards. When a thermal hazard happens, the system administrators and the facility manager must stop the production to avoid IT equipment damage and wear-out. In this paper, we study the thermal hazards signatures on a Tier-0 datacenter room's monitored data during a full year of production. We define a set of rules for detecting the thermal hazards based on the inlet and outlet temperature of all nodes of a room. We then propose a custom Temporal Convolutional Network (TCN) to predict the hazards in advance. The results show that our TCN can predict the thermal hazards with an F1-score of 0.98 for a randomly sampled test set. When causality is enforced between the training and validation set the F1-score drops to 0.74, demanding for an in-place online re-training of the network, which motivates further research in this context.**

***Index Terms*—HPC, Thermal Hazard, Predictive Model, Thermal Anomaly Detection, Temporal Convolutional Network**

I. Introduction and Related Work

The ICT sector's total electricity consumption is expected to reach 20% of the world-wide demand by 2030, with data centers expected to account for one-third of that [1]. Cooling is a high cost item for datacenter operation. The power-usage efficiency ratio (PUE) expresses the additional power required by the IT for removing the heat produced by the IT power consumption. While air-cooled datacenters easily reach PUE up to 2 [1], advances in cooling technologies like direct-liquid, hot water, and free-cooling can reduce it close to almost 1 [2]. In 2016, Google announced a PUE of 1.12 [3], while in 2018, NREL achieved the world-record PUE of 1.036 by leveraging thermosyphon technology [2]. A higher than nominal coolant temperature is required to leverage free-cooling in temperate regions [4], [5], which increases the risks of thermal runaway. In the scientific computing sector, in Europe, a EuroHPC pre-exascale system costs on average ∼600K€ per day[1]. Thus each day on which the supercomputer causes to the European taxpayer a loss of ∼600k€. Whereas in the business datacenter sector, in 2016, an Amazon.com web service shortage would have cost, on average, 15M$ of revenue lost [6].

A *thermal hazard* is a dramatic increase in node temperature, which can be triggered by *i)* failures in the cooling equipment (i.e. Computer Room Air Conditioning) or *ii)* failures in the monitoring and controlling of the cooling system; this can lead to the outage of the datacenter, with severe societal and business losses. Detecting thermal hazards in time is of extreme importance to avoid IT and facility equipment damage. Therefore, holistic monitoring systems are in place to monitor and visualize the datacenter state over time [7].

At the same time, progress in Deep Learning (DL) has enabled techniques for training models on large-scale time-series data. A recent DL architecture for detection of patterns in time series is the Temporal Convolutional Network (TCN) [8], which has proved able to outperform Long Short-Term Memory (LSTM) nets [9]. As such, TCNs are good candidates to perform prediction of thermal hazards, if a large data set of thermal events is available. For these reasons we choose a TCN model for our work. TCNs use dilated causal 1D-convolutions inside residual blocks. For sequence-to-sequence modeling, they can map the input series to an output with the same length. Since our TCN is a probability predictor, we equipped it with a block of dense layers at the end.

In the SoA, thermal hazards have been studied with different methodologies. [10] proposed to use simulators. [11], [12] proposed Machine Learning (ML) approaches, [13] proposed mathematical models, and finally, [14] proposed to use sensors with a computer model to create the room's heat map or thermal evolution model. While the simulator is hard to tune to the real environment, [11] used an Artificial neural network (ANN) model trained with offline simulation data of Computational Fluid Dynamics (CFD). Compared to the CFD, the ANN model's fast response time is suitable for an online predictor. To the best of our knowledge, no one has leveraged the large data available from holistic monitoring systems to study the statistical thermal hazard distribution, or proposed a data-driven Big Data (BD) and DL model for predicting thermal hazards.

In this work, *i)* we study the room thermal distribution during thermal hazards in a real Tier-0 datacenter; under these hazards conditions, we found regularity across many nodes with a significant percentage of them having an inlet temperature above the 95% quantile. *ii)* we propose a statistical approach to detect thermal hazards for an HPC room. With the proposed

This work has been partially supported by the EU H2020 ICT/2018 project IoTwins (g.a. 857191) and Emilia-Romagna POR-FESR 2014-2020 project "SUPER: SuperComputing Unifier Platform – Emilia-Romagna.

[1]The EuroHPC program has invested ∼650M€ in CAPEX ad OPEX for the three procured pre-exascale systems with an estimated daily average cost of ∼600k€ for a supercomputer - https://www.etp4hpc.eu/euexascale.html .

method, we obtained a dataset with 19.5% of hazard labels. *iii)* we investigate different machine and deep learning models (SVMs, SGD-classifier, LSTM, and TCN) in predicting the thermal hazard events 6 hours before they happen, which would give ample time for taking proactive countermeasures. Based on a set of experiments, we identify an optimal TCN achieving an F1-score of 0.98 in the hazard prediction for a randomly sampled. When causality is enforced between the training and validation set, the F1-score drops to 0.74, demanding for an in-place online re-training of the network, which motivates further research in this context.

II. Background Setup

Our study focuses on Marconi-A2 (KNL), the largest partition of the Tier-0 cluster Marconi at the CINECA datacenter, where it was hosted in the *Marconi KNL Room* (Figure 4, bottom-left). In this room Marconi-A2 was composed of 3312 nodes with one 68-cores Intel Xeon Phi 7250 CPU Knights Landing (KNL) running at $1.4\,\mathrm{GHz}$. Nodes had a $16\,\mathrm{GB/node}$ MCDRAM and a $96\,\mathrm{GB/node}$ DDR4. The internal network was Intel OmniPath Architecture 2:1. The cluster's peak performance was $11\,\mathrm{PFlop/s}$ [15].

Marconi KNL Room hosted 46 racks, plus 1 rack of switches, arranged in 3 rows; each rack had 18 stacked chassis, each with 4 nodes, totaling 3312 compute nodes. All racks had Rear Door Heat eXchangers. The room's 2 *hot aisles*, and 2 *cold aisles* were supported by 6 Computer Room Air Conditioning units.

CINECA runs a holistic monitoring framework, called EXAscale MONitoring (ExaMon) [7], scalable and capable of high-rate HPC telemetry from a wide range of heterogeneous sensors and data sources. For each cluster node and associated components, such as voltage regulators and fans, the Intelligent Platform Management Interface (IPMI) provides remote telemetry access to the built-in sensors [16]. ExaMon collects sensor data via IPMI at sampling interval $20\,\mathrm{s}$ [7] via an MQTT broker, and stores them using KairosDB, a specialized time-series database built on Cassandra (a NoSQL database management system), remotely accessible via RESTful APIs.

III. Thermal Hazard Prediction Methodology

In this section, we define the statistical tool for thermal hazard detection, based on the analysis of two real reported thermal emergencies. We use this characterization to generate ground-truth labeling of the HPC room for the whole year 2019. We then suggest a framework for thermal hazard prediction, which encompasses data query and preprocessing, model training, and final model inference, which provides the prediction.

A. Thermal Hazard Analysis and Labels Generation

Based on the study in [16], CINECA Marconi KNL Room had two known physical-thermal-hazard events in 2019: one on 28$^{\text{th}}$ June (peak from 16:00 to 19:00), and one on 1$^{\text{st}}$ July (peak from 14:30 to 17:00). In this paper, physical-thermal-hazard refers to these two recorded failures of the cooling system. We analyze the distribution of temperatures during these two peaks to compare the hazard distribution with the non-hazard distribution: the aim is to find indicators of the thermal hazards in the temperature data.

As a non-hazard distribution, we use the temperatures of the nodes in June and July, and downsampled to $1\,\mathrm{Sample/minute}$: this yields $\sim$ 88k samples, large enough to be representative of the ordinary temperature distribution. In this study,

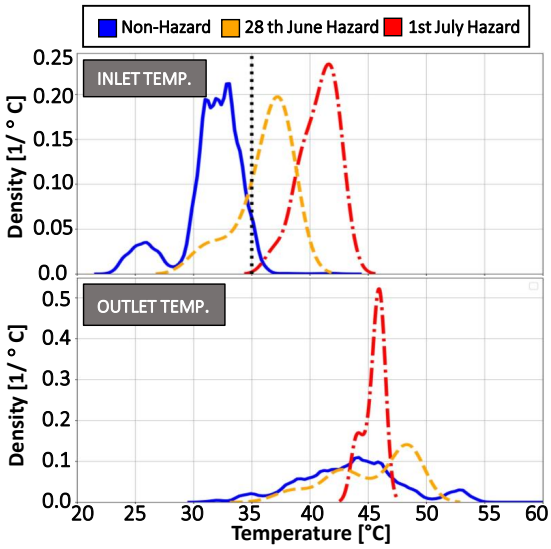

Fig. 1: Temperature distributions for Marconi A2's node 141 in June-July 2019.

	Time	Node_1	...	Node_3311	Node_3312
6 HOURS	2019-01-25 00:00:00	30°C	...	41°C	42°C
	...	...	...	...	...
	2019-01-25 05:58:00	29°C	...	39°C	40°C
	2019-01-25 05:59:00	28°C	...	39°C	40°C

(a) Inlet Temperature dataset

	Time	Node_1	...	Node_3311	Node_3312
6 HOURS	2019-01-25 00:00:00	FALSE	...	TRUE	TRUE
	...	...	...	...	...
	2019-01-25 05:58:00	FALSE	...	FALSE	FALSE
	2019-01-25 05:59:00	FALSE	...	FALSE	FALSE

(b) True-False table

Fig. 2: Time Windowing and Labeling.

we selected two temperature metrics from ExaMon: the inlet temperature `BB_Inlet_Temp` and the outlet temperature `Exit_Air_Temp`. These are the metrics most related to room temperature and were taken for every computing node.

Figure 1-top reports the inlet temperature distribution of one node for the three cases: non-hazard, 28$^{\text{th}}$ June hazard, and 1$^{\text{st}}$ July hazard. The dashed black line is the quantile 0.95 of the node's non-hazard distribution: as it is evident, the quantile 0.95 is a threshold that separates well the non-hazard and hazard temperatures. Figure 1-bottom reports the same information for the outlet temperature: hazard and non-hazard distributions overlap much more compared to inlet temperature, making it impossible to discriminate by thresholding on outlet temperature. We inspected some randomly selected nodes with this approach. We determined that the single-node quantile 0.95 of the non-hazard inlet temperature is a good parameter to discriminate between hazard and non-hazard.

1) Node-threshold

Based on the characterization of thermal hazards described above, we introduce a *node-threshold* defined for each node as the 0.95 quantile of its inlet temperature distribution over the entire dataset (which covers the whole 2019 year). Figure 2(a) summarizes a 6-hour time window (TW) of the inlet temperature dataset. We applied the node-threshold to assign to each $(\mathrm{node}, \mathrm{time})$ cell a `True`/`False` label indicating sample-by-sample thermal trouble, as shown in Figure 2(b). We empirically chose $\mathrm{TW} = 6\,\mathrm{hours}$.

2) Spatial-temporal-impact-threshold

To assign hazard /non-hazard labels to TWs (Figure 2(a)), not just to samples, we exploit the time-series nature of the temperature data. We introduce a *spatial-temporal-impact-threshold*

able to account for thermal hazards' spatial and temporal continuity. A 3312-node 6-hour TW with $1\,\mathrm{Sample/minute}$ amounts to $3312 \times 6 \times 60 = 1192320$ `True`/`False` values (Figure 2(b)). The *spatial-temporal-impact-threshold* regulates the portion of `True`'s inside the TW required to declare the room in thermal hazard. A higher quorum will select thermal hazards that are more widespread, i.e. involve more nodes for a longer time.

It is essential to remark that, though based on real information extracted from the physical hazard distribution, this statistical labeling approach is artificial and must be confirmed by comparing with the ground-truth reported thermal emergencies. As made evident in Figure 3(x-axis is date), if we set the *spatial-temporal-impact-threshold* to 5%, our statistical approach captures the reported ground-truth thermal emergency, while detecting additional thermal hazards, which were unnoticed by the system administrators. Indeed, these are conditions for which the compute nodes' temperatures have drastically increased without causing immediate damage, but still possibly damaging the nodes. With our statistical labeling approach, we can capture these events which are unnoticed by humans.

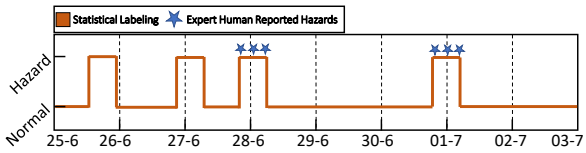

Fig. 3: Thermal Hazard Detection

If we increased the *spatial-temporal-impact-threshold* quorum to 25%, the statistical labeling approach could only detect the second hazard, thus being to restrictive in identifying unnormal states. For the selected *spatial-temporal-impact-threshold* $= 5\%$, the room is labeled in thermal hazard for 19.5% of the time in 2019.

B. Thermal Hazard Prediction Framework

Our thermal hazard predictor is a model that, based on time series data of computing nodes' sensors, predicts if a thermal hazard will happen in the room in the next hours. Input data are the time series of nodes' temperature (and power consumption), and the output is a binary classification: likely forthcoming hazard or not. We define *Prediction Horizon* (PH) the label's time distance since the last input data. For instance, if the input is the temperature TW 00:00:00-5:59:59 and PH $= 6\,\mathrm{hours}$, the task is to predict the state of the time interval 06:00:00-11:59:59. For the PH $= 0$, we have the *detection* task.

In particular, PH $= 6\,\mathrm{hours}$ was chosen upon discussions with system administrators, as a tradeoff sufficient to provide enough time for the different correction actions to be taken by the system administrators. Treating PH $= 6\,\mathrm{hours}$ as a time lag, the hazard/non-hazard binary ground-truth labels have autocorrelation 0.65 over the year 2019 and identifying ground-truth labels 6 hours apart as the output and target of a Last-Value Predictor (LVP) yield an F1-score of 0.72. Being the LVP, the simplest (non-)model, F1 $= 0.72$ is a baseline any proposed model must be compared against.

Figure 4 illustrates our proposed architecture for the thermal hazard predictor, composed of three main components: the architecture for data collection, storage, based on ExaMon (section II), the thermal hazard analysis including the data extraction, preprocessing (e.g., missed data handling, time alignments), label generator and data loader, and the Deep Learning (DL)-powered thermal hazard prediction system (training and inference). For the DL model used for prediction, a Temporal Convolutional Network (TCN) is selected [8].

The TCN's input is a TW of data extracted from the database. In the off-line training stage, a large set of TWs is extracted (training set), and preprocessed to generate the ground-truth labels with the two-threshold statistical approach of Section III-A. Inferences with the trained model are the predictions of thermal hazards.

Relying on ExaMon, it is possible to implement and test DL models using a very broad set of node metrics collected from sensors: the database stores hundreds of metrics, of which 42 are IPMI metrics. In this work, we focused on the nodes' inlet temperature (as motivated in Section III-A), and we plan to boost our TCN by adding the nodes' power consumption.

IV. Preliminary Results

In this section we evaluate the Temporal Convolutional Network in predicting thermal hazard in CINECA's Marconi KNL Room. We describe the dataset, introduce our TCN topology and competitor models, and finally discuss two experiments highlighting our TCN's promising prediction skills.

A. Experimental Dataset

Experiments were based on the inlet temperature time series of HPC room, which hosts 46 racks containing 18 chassis, each chassis include 4 nodes. So in total sensory data of 3312 nodes, for the whole year 2019. There were two thermal hazards on 28$^{\text{th}}$ June and 1$^{\text{st}}$ July. We have utilized the IPMI interface to data collection with a sampling rate of 20 seconds; then, data were downsampled to 1 minute in the preprocessing step.

We generated the ground-truth labels with the statistical approach described in Section III-A, with *node-threshold* $= 0.95$ and *spatial-temporal-impact-threshold* $= 0.05$ as motivated. With these values, 19.5% of the data is labeled as a thermal hazard, sufficient for training our algorithms.

B. TCN and competitor predictors

The proposed TCN has 2 blocks: (1) a *Feature Learning Block* (14k parameters) of 7 1D-convolutional layers with average pooling; (2) *Classification Block* (173 parameters) of 4 dense layers of 15, 6, 4, 3, 2 units. All layers present the batch normalization and the ReLU activation. We compare our TCN against other models[2]:

0) Last Value Predictor (LVP): minimum baseline for any time-series task; the prediction $\hat{y}$ is simply a copy of the present observation y_{true}: $\hat{y}(t + \text{PH}) = y_{\text{true}}(t)$, with PH prediction horizon as defined in Section III-B.

1) Support Vector Machine (SVM): SVM with either linear or Radial Basis Function (RBF) kernels. SVMs produce decision boundaries with margins to improve generalization.

2) Stochastic Gradient Descent (SGD)-classifier: linear SVM trained with SGD instead of convex optimization, enabling larger train set size.

3) Long Short-Term Memory (LSTM): a type of Recurrent Neural Network (RNN) that learns long-term dependencies

[2]SVMs and SGD-classifier were implemented in Scikit-learn 0.23; LSTM was implemented in Keras 2.4; TCN was implemented in PyTorch 1.5.

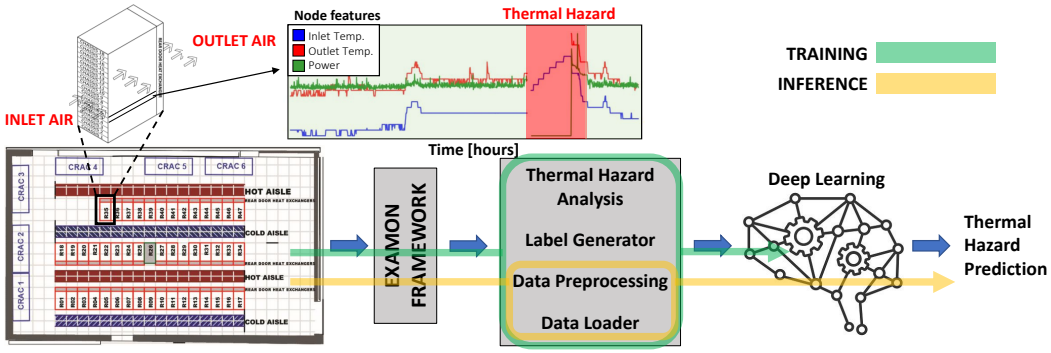

Fig. 4: Architecture for Thermal Hazard Predictor.

TABLE I: Prediction Results

	Recall	Precision	F1-score
Experiment 1: random validation set			
Last value predictor	0.72	0.72	0.72
Linear SVM	0.55	0.56	0.55
RBF-SVM	0.80	0.94	0.86
SGD-classifier	0.64	0.76	0.69
LSTM	0.84	0.98	0.91
TCN - this work	**0.97**	**0.99**	**0.98**
Experiment 2: time-separate validation set			
TCN - this work	**0.79**	0.70	**0.74**

thanks to additional gates [9]. Our LSTM has 2 layers of hidden and output size 16, followed by a dense layer.

To keep the parameter space of the models small, the models were built using as input only the `BB_Inlet_Temp` temperatures of 72 nodes which composes one rack. The rack was selected randomly in the room. We remark that all the 3312 nodes were used for generating the thermal hazard labels.

C. Experiment 1: random validation set.

In the random validation set experiment, we selected the validation set randomly as 20% of the 2019 data, and trained all models on the remaining 80%. Table I shows the results.

The linear SVM yields F1-score 0.55, essentially random and worse than the LVP-baseline: this is due to the linear models' poorness and to the train set reduction made necessary by computational complexity. The RBF ranks better, with F1-score 0.86, which is also 0.17 above the SGD-classifier. Both DL models outperform the non-deep ones: the LSTM reaches F1-score 0.91, and our TCN ranks best, with F1-score 0.98.

D. Experiment 2: time-separate validation set

To simulate a real case scenario, we trained the TCN with only May 2019 data and validated the model in the first week of June 2019. Our TCN achieved an F1-score of 0.74 F1-score, which is 0.24 lower than Experiment 1 with a random validation set. Such degradation is due to the random selection of the validation set in Experiment 1 for which similar samples are present in the training and validation set. Experiment 2 is, however, closer to the real usage of the predictive model. We suspect that the limited accuracy of Experiment 2 is caused by (1) the limited set of nodes considered for the prediction, and (2) a non-stationarity in the thermal effects that is not captured if we use only past data to predict the future. These will be investigated in future works.

V. Conclusion

In this study, we used statistical analysis of real thermal hazard data from CINECA Marconi KNL Room to characterize thermal hazards and proposed a thermal hazard predictor, namely a Temporal Convolutional Network, which outperforms non-deep models and LSTM. Our TCN has a 0.24 drop in F1-score when applied in a scenario simulating a real case of training limited to (recent) past data. In future work, we aim to improve the results by different strategies: (*i*) training on more historical data; (*ii*) addition of input metrics, prioritizing power consumption; (*iii*) 2D and 3D-convolutions; (*iv*) iterative retraining, to simulate the real scenario even more accurately.

References

[1] N. Jones, "How to stop data centres from gobbling up the world's electricity," *Nature*, vol. 561, no. 7722, pp. 163–167, 2018.

[2] NREL. [Online]. Available: nrel.gov/computational-science/measuring-efficiency-pue.html

[3] J. Gao *et al.*, "Machine learning applications for data center optimization," 2014.

[4] H. Shoukourian *et al.*, "Forecasting power-efficiency related key performance indicators for modern data centers using LSTMs," *Future Generation Computer Systems*, vol. 112, pp. 362–382, Nov. 2020.

[5] C. Conficoni *et al.*, "Hpc cooling: A flexible modeling tool for effective design and management," *IEEE Transactions on Sustainable Computing*, 2018.

[6] J. L. Hennessy *et al.*, *Computer Architecture, Sixth Edition: A Quantitative Approach*, 6th ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2017.

[7] A. Bartolini *et al.*, "Paving the way toward energy-aware and automated datacentre," in *Proceedings of the 48th International Conference on Parallel Processing: Workshops*, ser. ICPP 2019. New York, NY, USA: ACM, 2019, pp. 8:1–8:8.

[8] S. Bai *et al.*, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv:1803.01271*, 2018.

[9] S. Hochreiter *et al.*, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] J. Cho *et al.*, "Measurements and predictions of the air distribution systems in high compute density (internet) data centers," *Energy and buildings*, vol. 41, no. 10, pp. 1107–1115, 2009.

[11] J. Athavale *et al.*, "Artificial neural network based prediction of temperature and flow profile in data centers," in *2018 17th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*. IEEE, 2018, pp. 871–880.

[12] M. Marwah *et al.*, "Thermal anomaly prediction in data centers," in *2010 12th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, 2010, pp. 1–7.

[13] L. Wang *et al.*, "Towards thermal aware workload scheduling in a data center," in *2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks*. IEEE, 2009, pp. 116–122.

[14] Q. Tang *et al.*, "Sensor-based fast thermal evaluation model for energy efficient high-performance datacenters," in *International Conference on Intelligent Sensing and Information Processing*. IEEE, 2006.

[15] E. Rossi. (2017) Marconi-a2 (knl). [Online]. Available: http://www.hpc.cineca.it/hardware/marconi

[16] M. Seyedkazemi Ardebili *et al.*, "Thermal characterization of a tier0 datacenter room in normal and thermal emergency conditions," in *Proceedings of High Performance Computing in Science and Engineering 2019*.