ANL/NSE-21/80

Physics-Infused AI/ML Based Digital-Twin Framework for Flow-Induced-Vibration Damage Prediction in a Nuclear Reactor Heat Exchanger

Updates on Implementation of a NoSQL Database and Multi-Time-Series Thermal State Predictions at Thousands of 3D Point-clouds

Nuclear Science and Engineering Division

About Argonne National Laboratory
Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

Disclaimer
This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor UChicago Argonne, LLC, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, Argonne National Laboratory, or UChicago Argonne, LLC.

ANL/NSE-21/80

Physics-Infused AI/ML Based Digital-Twin Framework for Flow-Induced-Vibration Damage Prediction in a Nuclear Reactor Heat Exchanger

Updates on Implementation of a NoSQL Database and Multi-Time-Series Thermal State Predictions at Thousands of 3D Point-clouds

prepared by
Subhasish Mohanty and Richard B. Vilim
Nuclear Science and Engineering Division, Argonne National Laboratory

Prepared for
United State Department of Energy, Advanced Nuclear Technology Development Program

September 30, 2021

ABSTRACT

This report summarizes some of the ongoing work related to the development of an expert-elicitation-digital-twin framework for real time damage state prediction in heat exchanger components of a nuclear reactor. The framework is targeted towards predicting damage associated with coupled low cycle fatigue (associated with regular heat-up, cool-down and power operation transients) and high cycle fatigue (associated with flow induced vibration transients). The overall framework will be based on a NoSQL based database, physics-infused-geometry-dependent virtual-sensor data, different AI/ML techniques-based data-driven-predictive-model applications (Apps) and real-time plant sensor measurements available through few existing sensors. Towards this overall goal, this report updates some of the ongoing work, such as on implementation of a NoSQL Database (such as MongoDB), FE based heat transfer analysis of a heat exchanger (e.g. of a PWR steam generator) for generating geometry-dependent virtual sensor data and evaluation of various AI/ML models such as based on multivariate linear regression, ensembled decision-tree based Random-Forest and Gradient-Boosting regression and high-dimensional-kernel-function-transformation based Support-Vector-Machine regression models. The AI/ML models were evaluated for predicting multi-time-series thermal states at thousands of 3D point-clouds.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

AI	Artificial Intelligence
DB	Design Basis
DT	Digital Twin
GLF	Grid Load Following
GB	Gradient Boosting
HE	Heat Exchanger
ML	Machine Learning
RF	Random Forest
SVM	Support Vector Machine

ACKNOWLEDGMENTS

The authors acknowledge the support of United State Department of Energy's award under the U.S. Industry Opportunities for Advanced Nuclear Technology Development Program. The authors also acknowledge the help of Dr. Sasan Bakhtiari for some of the NDE related data presented in this report.

1 Introduction

1.1 Different Causes Those Affect Flow-Induced-Vibration Damage in Heat-Exchanger Tubes

The Flow-Induced-Vibration (FIV) damage in a heat exchanger (HE) tube is not just due to flow induced vibration rather due to coupled effect of many variables. Figure 1. 1 shows the schematic showing the different inter-linked causes those affect the overall FIV damage in HE components such as in tubes and tube-support baffle plates.

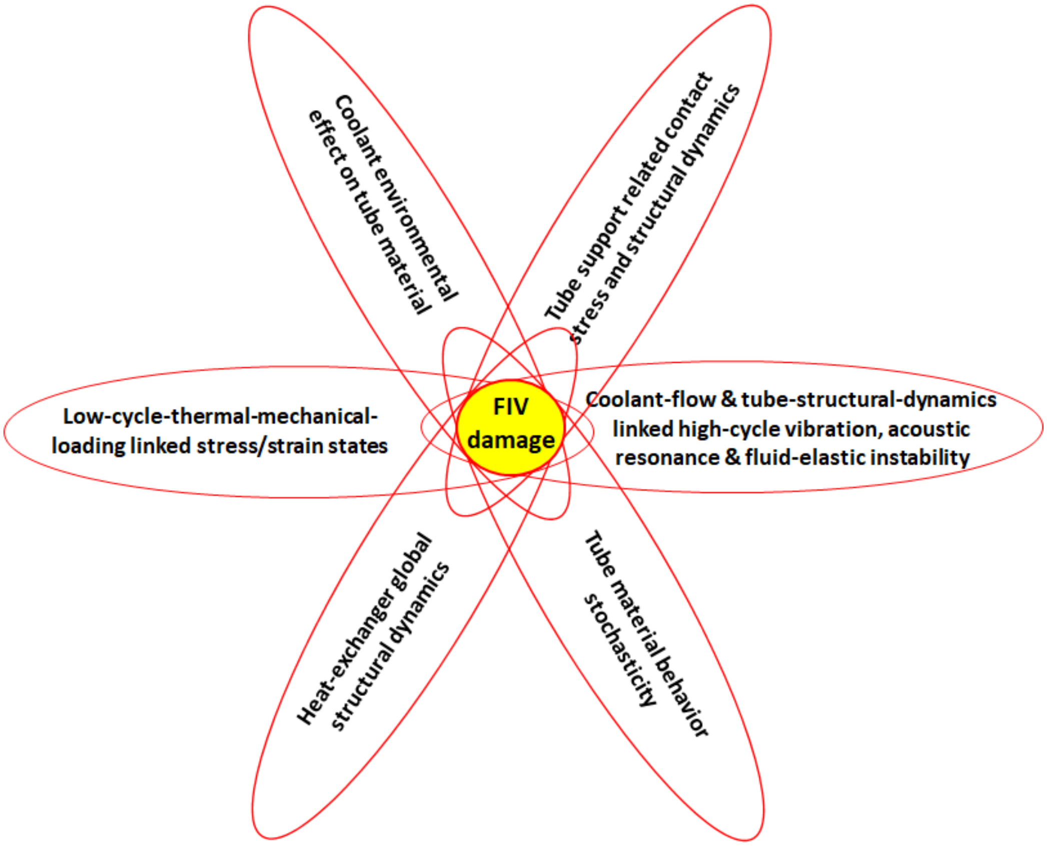

Coolant environmental effect on tube material
Tube support related contact stress and structural dynamics
Low-cycle-thermal-mechanical-loading linked stress/strain states
FIV damage
Coolant-flow & tube-structural-dynamics linked high-cycle vibration, acoustic resonance & fluid-elastic instability
Heat-exchanger global structural dynamics
Tube material behavior stochasticity

Figure 1. 1 Schematic showing the different inter-linked causes those affect the overall FIV damage in a HE component.

Although FIV is one of the major reasons of tube failure in a HE, it doesn't occur alone as a single failure mode. Rather the related high-cycle fatigue damage is coupled with the low-cycle fatigue damage modes (associated with the reactor thermal-mechanical loading cycles). When a tube is subjected to thermal-mechanical loading cycles, some tubes may undergo high stresses and related local plastic deformation and bowing. The plastic deformation and blowing can be due to thermal stratification along the length of the tube. The local plastic deformation can be exaggerated by FIV (associated with high-frequency oscillations and related acoustic resonance and fluid-elastic instability). The plastic deformation can be further exaggerated if the tubes rub against the tube supports. Essentially the tubes are damaged due to the combined effect of low-cycle and high-cycle loading transients. Figure 1.2 shows the schematic of an embedded high-cycle-fatigue (HCF) loading (associated with flow-induced-vibration loading) in a low-cycle-fatigue (LCF) cycle (associated with the regular thermal-mechanical loading cycles). The LCF loading cycles can alone lead to substantial thermal strain that in turn can produce substantial stress and plastic deformation depending on the thermal-mechanical boundary conditions. Figure 1.3 shows the example increase of thermal strain in a stainless-steel specimen during heating up. Depending on the material types, similar thermal strains are expected in actual reactor component subjected to similar temperature ramp. Additional strain and associated stress and plastic deformation can be due to FIV related high-frequency oscillations. This combined effect of LCF and HCF may be a reason for much reduced life of HE tubes than that alone by HCF or LCF. However, it's difficult to segregate HCF damage from LCF damage because they are coupled and must be modeled combinedly rather than separately. In top of LCF and HCF related effects, the damage growth in a tube can be dependent on other factors such as on coolant environmental effect on tube materials and on geometric/support boundary conditions of tubes. Overall FIV damage in a tube are the coupled effect of the followings:

a) LCF loading: Regular thermal-mechanical-loading cycle linked stress/strain states.

b) HCF loading: Coolant-flow & tube-structural-dynamics linked high-cycle vibration, acoustic resonance & fluid-elastic instability related additional stress/strain states (in top of LCF related stress/strain states).

c) Tube support related contact stress and structural dynamics: Under combined effect of LCF and HCF a tube can bow and rub against a support/baffle plate or against adjacent tubes and can result in tube fretting wear. The fretting wear lead not only to a complex nonlinear contact stress (at tube-to-tube and tube-to-baffle contact locations), but also can lead to a potential site of fatigue crack initiation.

d) HE global structural dynamics: The high-cycle local vibration level in a tube can also be affected by the global structural dynamics of a HE. The global structural dynamics can be due to imbalance flow associated with tube plugging, differential bowing of tubes, etc.

e) Coolant environmental effect on tube material: In top of the above mentioned causes the coolant environmental effect can significantly affect the performance of the tube material. The environmental effect can be function of fluid/metal temperature, concentration of dissolved oxygen (DO) in the coolant water, and the stress/strain amplitude associated with combined HCF and LCF loading cycles.

f) Tube material behavior stochasticity: Damage initiation and progression in a tube are not necessarily deterministic rather stochastic. This stochastic nature of damage initiation and progression is associated with variation in material microstructure, surface finish, etc. Hence, the variation of material behavior also needs to be modeled e.g., using some sort of probabilistic methods.

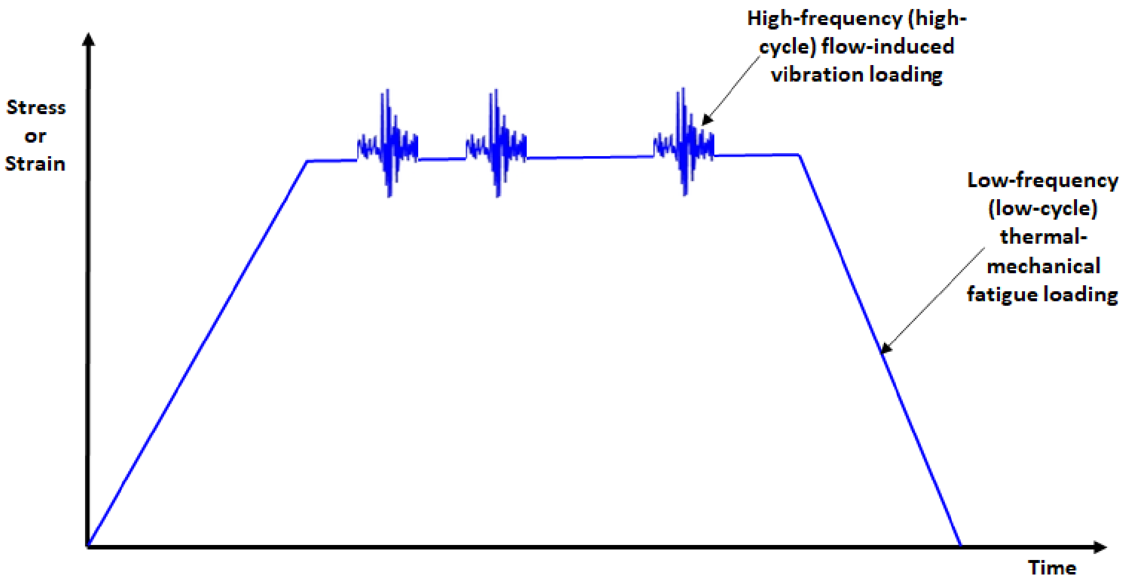

High-frequency (high-cycle) flow-induced vibration loading
Stress or Strain
Low-frequency (low-cycle) thermal-mechanical fatigue loading
Time

Figure 1. 2 Schematic showing the embedded high-cycle-fatigue cycle (associated with flow-induced-vibration transients) in a low-cycle-fatigue cycle (associated with the regular thermal-mechanical transients during heat-up, cool-down and power operation).

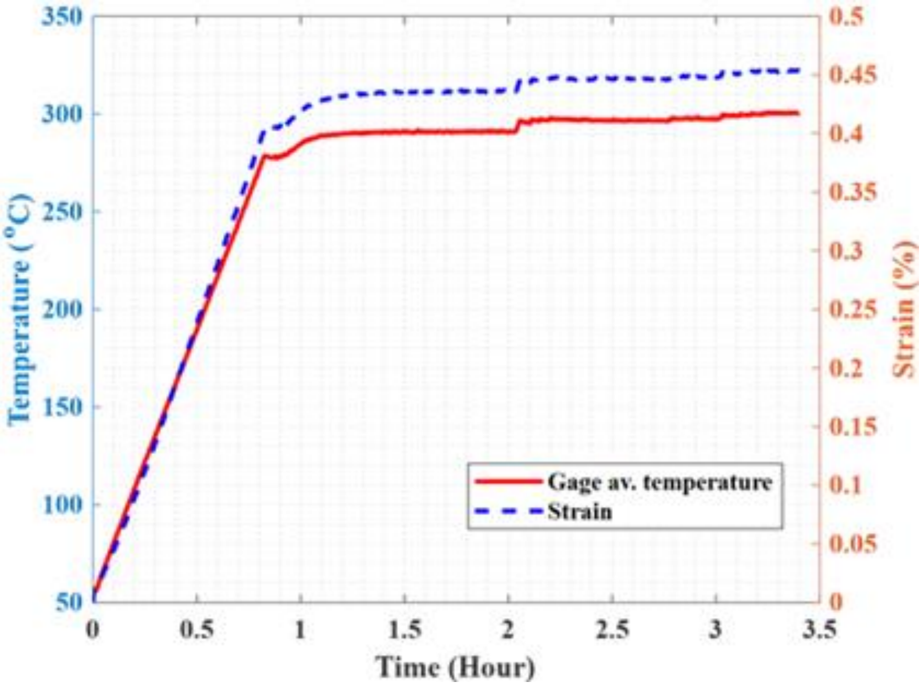


Figure 1. 3 Example increase of thermal strain in the gage area of a stainless-steel specimen during a heating up experiment [1].

1.2 Need of an Online Monitoring Based Digital-Twin System for Predicting FIV Damage

Currently, nuclear power is a major source of carbon-free electricity in the United States. However, sustaining the nuclear reactor fleet operation is a challenge due to ever increasing cost associated with maintenance of an ageing nuclear reactor fleet. When the reactors will be operated for much longer time (say 80 or 100 years) compared to the original design life of 40 years, more cracking of component is expected due to the long-term exposure of reactor material to the harsh reactor-coolant environment. That means some of the safety-critical components require more frequent inspections of the components. This is in turn requires frequent shutting down of the NPP for NDE inspection. However, the economic loss associated with frequent shutting down related power outage and additional NDE inspection related cost can negatively impact the overall economic performance of NPP operators. For this reason, NPP operators are increasingly looking for online monitoring (OLM) of plant components. However, one of the major drawbacks of the OLM based plant inspection technique is that it requires to place many sensors at different locations of the component for accurately estimating the state of that component. In an NPP, it is not that easy to place sensors at the required locations. This is because of the survival of the sensors in harsh radiation environment, difficulty in making new port for insertion of sensor and its instrumentation, stringent regulatory requirement of qualifying any sensor for NPP use, etc.

The advancement of artificial intelligence and machine learning (AI-ML) techniques hold promise in use of an OLM based plant inspection system. This is through use of upcoming digital-twin (DT) technology. The DT based OLM models can be considered as a live model of an actual reactor 3D-components, an assembly of components, or the entire nuclear power plant (NPP). **The DT model is a combination of online monitoring, AI-ML and physics based models of the actual components. This DT model can be used for expert elicitation of spatial-temporal component states (at thousands to millions of 3D point-clouds), even there are few online measurement sources available (such as from a few existing plant sensors currently being used for plant thermohydraulic or process measurements)**. The DT framework can continually adapt to the operational changes based on the collected offline and online data and can be used for forecasting the future state of a component at any given time. This will help with migrating from the current reactive/periodic inspection-maintenance regime to a proactive/predictive inspection-maintenance regime. In turn, it will help increase the availability and efficiency of a component, extend its useful life cycle, and reduce its inspection-maintenance cost.

1.3 Major Components of a Digital-Twin System

A DT system has to be built on an AI-ML based data-driven analytics framework. Figure 1.4 shows the schematic diagram for the flow of data in the proposed DT framework, which is currently under development. Different subcomponents are briefly described below.

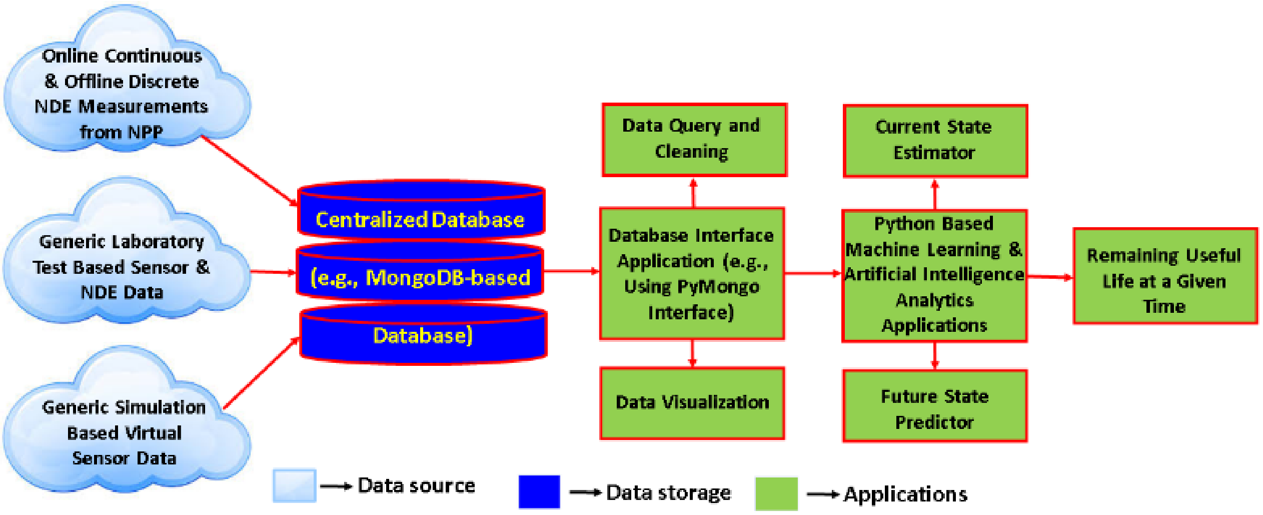

Online Continuous & Offline Discrete NDE Measurements from NPP
Generic Laboratory Test Based Sensor & NDE Data
Generic Simulation Based Virtual Sensor Data
Centralized Database
(e.g., MongoDB-based
Database)
Data Query and Cleaning
Database Interface Application (e.g., Using PyMongo Interface)
Data Visualization
Current State Estimator
Python Based Machine Learning & Artificial Intelligence Analytics Applications
Future State Predictor
Remaining Useful Life at a Given Time
→ Data source
→ Data storage
→ Applications

Figure 1. 4 Schematic showing the flow of data in the proposed DT framework.

a) **Historical data**: Since a DT model will be primarily based on AI-ML data-driven approach, this requires training or historical data to estimate the model parameters. The training data can be from physics based virtual sensor measurements (from multi-physics digital model of the actual physical asset) and real sensor measurements. The sensor measurements are either continuous measurements from an online monitoring (OLM) system (either through small scale laboratory tests and/or from actual plant sensor). Historical data can also include non-continuous measurements from offline in-service inspections that employ Non-Destructive Evaluation (NDE) methods.

b) **Real-time process measurements**: The DT model need to continuously predict the health of a component requiring updating the underlying process variables/parameters that affect the health of a component.

c) **Database**: The overdependence of a DT system on data (from virtual and/or real sensor measurements) necessitates access to a suitable database system in addition to the development of core analytics techniques and generation of underlying data. The database must store thousands to millions of time-series sensor measurements (either from virtual or real sensor measurements), with structured/un-structured data source, with different time scales (some measurements in the scale of milliseconds, some in the scale of hours to days), with continuous/non-continuous data source and at the same time work in a real time and agile environment. This type of multi-faceted requirements needs a suitable storage solution or database that must be efficient in providing big data storage, high scalability, high concurrent reading with low latency and high availability.

d) **Physics-infused hybrid analytics AI/ML applications (App)**: The DT system need to be dependent on different predictive models and associated software. For example, life of a component depends on different factors such as spatial-temporal variation of temperature, strain, stress, etc. To reduce complexity some of these underlying factors can have separate predictive model applications (Apps) or software. Each Apps can be separately developed and updated (with new features). Eventually the information from different Apps needs to be combined (in real time) for estimating the state of a component at any given time.

e) **Application programming interface (API)**: The DT model can have different database and predictive model applications. Suitable application programming interface (API) is required such that seamless flow of information can happen between different database and predictive model applications.

1.4 Physics-Infused hybrid analytics AI/ML framework for FIV Damage Prediction

The DT system for predicting FIV associated damage in real-time, need to be built on an AI-ML based data-driven analytics framework. This is for on-demand prediction of HE component structural states hence by improving the overall economics and safety of NPP. However, since the FIV damage in HE is geometry and various physics dependent, the AI/ML based data-driven analytics framework must be inspired through conventional physics models. Different physics infused Apps those can feed to the overall physics-augmented analytics framework need to be developed separately and to be combined through different APIs. Figure 1.5 shows the schematic showing the envisioned different physics infused AI/ML based Apps. For accurate predictions of FIV in a HE component all these separate Apps need to be developed and combined. However, depending on the funding and program priority we will be only developing few important Apps. For example, first our goal is to develop the framework for predicting the FIV related crack initiation life, then the propagation life. Towards this goal, in this report we present a physics-infused AI/ML based heat-transfer modeling methodology. This is for predicting process measurements-dependent spatial-temporal-temperature at any given time and locations of a HE component. Corresponding App is highlighted in Figure 1.5.

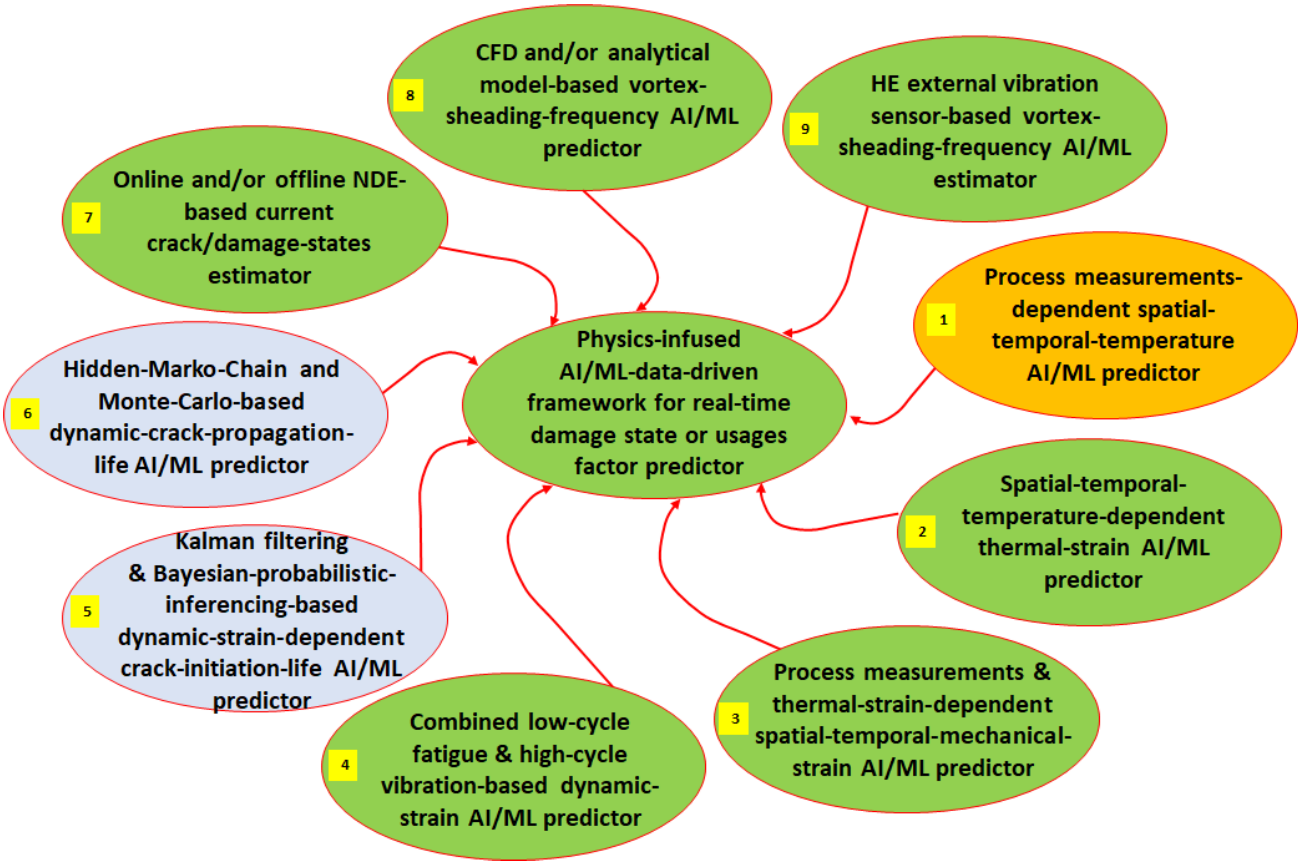

8
CFD and/or analytical model-based vortex-sheading-frequency AI/ML predictor
9
HE external vibration sensor-based vortex-sheading-frequency AI/ML estimator
7
Online and/or offline NDE-based current crack/damage-states estimator
1
Process measurements-dependent spatial-temporal-temperature AI/ML predictor
Physics-infused AI/ML-data-driven framework for real-time damage state or usages factor predictor
6
Hidden-Marko-Chain and Monte-Carlo-based dynamic-crack-propagation-life AI/ML predictor
2
Spatial-temporal-temperature-dependent thermal-strain AI/ML predictor
5
Kalman filtering & Bayesian-probabilistic-inferencing-based dynamic-strain-dependent crack-initiation-life AI/ML predictor
3
Process measurements & thermal-strain-dependent spatial-temporal-mechanical-strain AI/ML predictor
4
Combined low-cycle fatigue & high-cycle vibration-based dynamic-strain AI/ML predictor

Figure 1. 5 Schematic showing the different AI/ML based Apps those can feed to the overall physics-augmented analytics framework (Note: Results related to the highlighted App-1 are presented in this report. Additionally, our ongoing work [2-4] on other AI/ML related projects e.g., related to App-5 and 6 will feed to the overall project).

1.5 Organization of This Report

We are developing the DT framework for real-time FIV damage predictions in a HE component. The work is ongoing. In this FY 21 report we present some of the results related to the ongoing works. Below are the summary of the works presented in different sections of this report:

a) ***Section 2 Finite Element Analysis Based Heat Transfer Analysis for Generating the Spatial-Temporal Virtual Sensor Data:*** This section discusses details of the finite element (FE) based heat transfer analysis model of an example heat exchanger such as of a PWR steam generator. Based on the resulting FE model data the AI/ML models were trained and validated.

b) ***Section 3 MongoDB Based NoSQL Database Implementation:*** In this section, a summary of major differences between SQL and NoSQL type database and the reason for choosing a NoSQL database for nuclear reactor DT applications are discussed. Also, some of the example results related to implementation of NoSQL based database such as using commercially available MongoDB database are discussed.

c) ***Section 4 Theoretical Background of Different AI/ML Predictive Models:*** In this section, the theoretical background behind various AI/ML techniques such as multivariate linear regression, ensembled decision-tree based Random-Forest (RF) and Gradient-Boosting (GB) regression and high-dimensional-kernel-function-transformation based Support-Vector-Machine (SVM) regression models are discussed.

d) ***Section 5 Multi-Time-Series Temperature Prediction in a SG-Tube: Evaluation of Different AI/ML Models with Few Outputs Under a Design-Basis Loading Cycle:*** In this section, different AI/ML models described in section 4 were quantitatively evaluated for their accuracy considering a design-basis type loading cycle. The quantitative model evaluations were performed by predicting the temperature at only few locations.

e) ***Section 6 Multi-Time-Series Temperature Prediction in a SG-Tube: Predictions at Thousands of 3D Point-clouds Under a Design-Basis Loading Cycle:*** In this section, different AI/ML models described in section 4 were qualitatively evaluated for their accuracy. The qualitative model evaluations were performed by predicting the time-dependent temperature at thousands of 3D point-clouds of a SG-tube subjected to a Design-Basis (DB) loading cycle.

f) ***Section 7 Multi-Time-Series Temperature Prediction in a SG-Tube: Predictions at Thousands of 3D Point-clouds Under a Grid-Load-Following Loading Cycle:*** In this section, different AI/ML models described in section 4 were qualitatively evaluated for their accuracy. The qualitative model evaluations were performed by predicting the time-dependent temperature at thousands of 3D point-clouds of a SG-Tube subjected to a Grid-Load-Following (GLF) loading cycle.

2 Finite Element Analysis Based Heat Transfer Analysis for Generating the Spatial-Temporal Virtual Sensor Data

Real time prediction of structural damage (such as due to flow-induced-vibration) in a heat exchanger component, requires the knowledge of spatial-temporal distribution of stress and strain in that component. This is in turn requires the knowledge of spatial-temporal distribution of temperature. This includes the time-dependent temperature gradient along all the three coordinates of a component. The spatial-temporal temperature information is required at many locations to train an AI/ML based DT model. Although it is ideally possible to train an DT model using data from real sensors, however, it is not that easy to place sensors at all the required locations. This is because of the inaccessibility (e.g., inside thickness of a thicker component with substantial thermal gradient along the thickness of that component), survival of the sensors in harsh radiation environment, difficulty in making new port for insertion of sensor and its instrumentation, stringent regulatory requirement of qualifying any sensor for NPP use, etc. In that case a finite element (FE) based model can be developed for estimating the initial (or one time) spatial-temporal training data. These data can be referred as virtual sensor data. Once the DT model trained, the resulting model can be used for predicting the spatial-temporal states (e.g., the temperature, stress, strain, etc.) for a given real-time input variables or process measurements those affect those spatial-temporal states. In this report we present the DT model with only capability of predicting the spatial-temporal temperatures. The related training and validation data were generated through FE based heat transfer analysis models. Two heat transfer analysis model were developed: one under design-basis loading cycle and other under grid-load-following loading cycle. The details of the model are discussed below.

2.1 FE Model

2.1.1 FE CAD Model and Mesh

In the reported work we considered a steam generator (SG) that is primarily used in a pressurized water reactor (PWR) as an example component/system. The overall developed methodology (for developing the discussed DT framework) can be easily extended for other type of heat exchangers such as used for boiling water reactor (BWR). Figure 2.1 shows the CAD model of a PWR steam generator. Whereas Figure 2.2 shows the corresponding FE mesh. To note that for the demonstration purpose, we only considered a single U-bend tube that connect the hot-leg and cold-leg side of the SG. The developed methodology can be repeated for all the tubes in SG. Figure 2.3 shows the FE mesh of the steam generator tube.

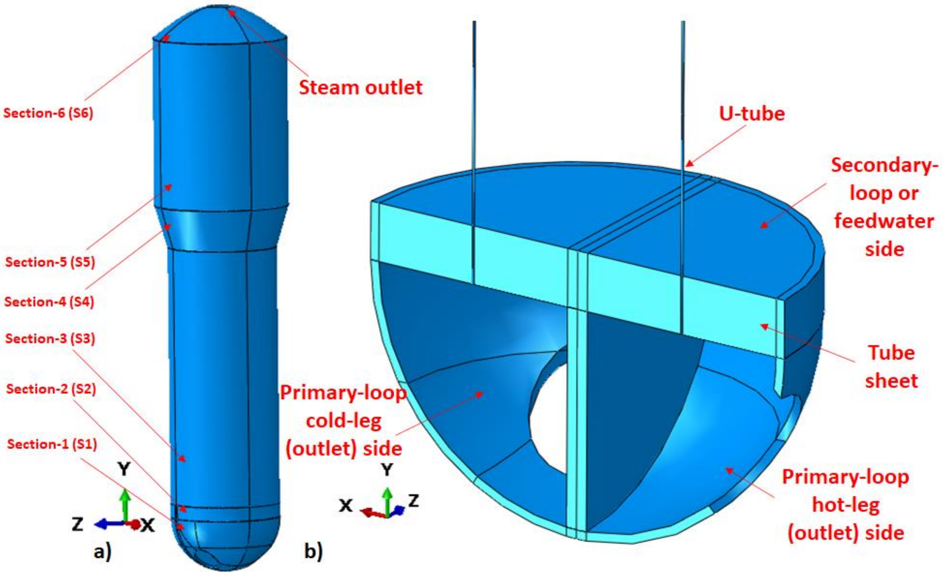


Figure 2. 1 a) CAD model of a PWR steam generator b) Cross-sectional view of the CAD model near the bottom head of steam generator.

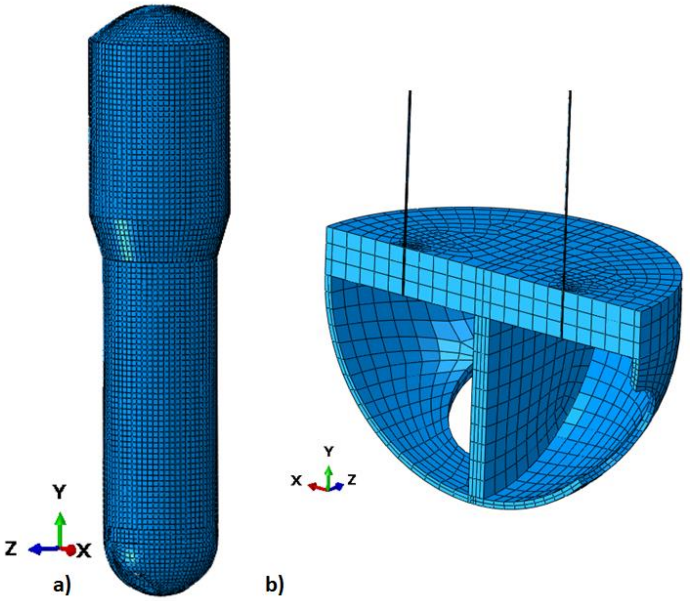


Figure 2. 2 a) FE mesh of the PWR steam generator b) Cross-sectional view of the FE mesh near the bottom head of steam generator.

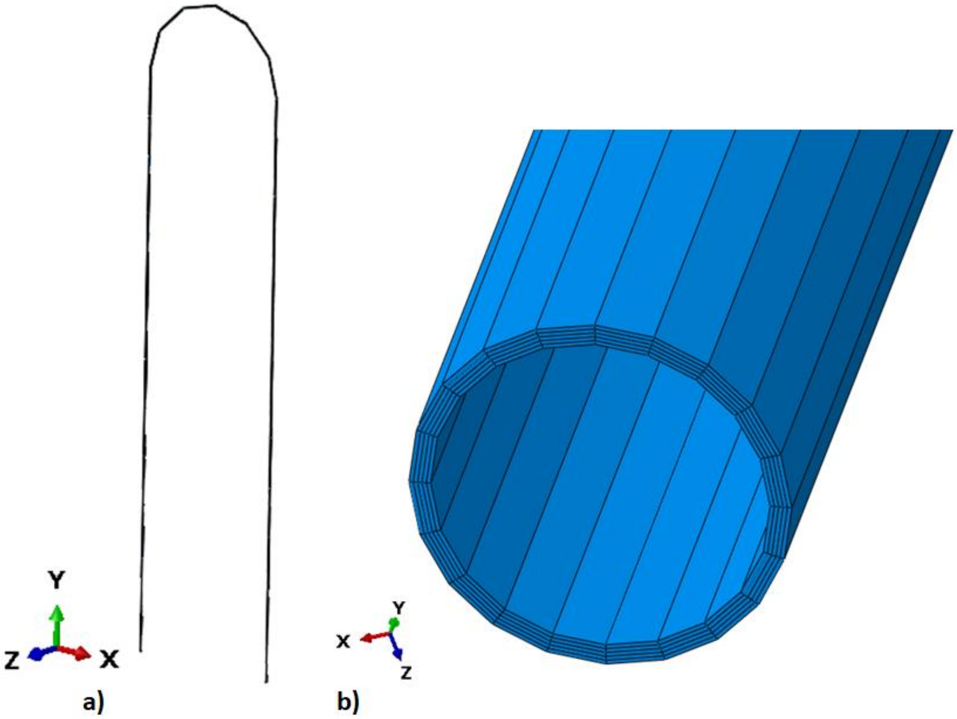


Figure 2. 3 a) FE mesh of the steam generator tube b) Cross-sectional view of the FE mesh near one end of the tube.

2.1.2 ASME Code Based Input Material Properties

In the reported FE model, we assumed SA508 low alloy steel and IN600 Nickel alloy material for SG shell and tube, respectively. We used the ASME code [5] based respective thermal properties for developing the heat transfer analysis models. Figure 2.4 shows the temperature-dependent thermal conductivity for low alloy steel (SA508) and Nickel alloy (IN 600). Whereas Figures 2.5 and 2.6 show the corresponding specific heat capacity and mean coefficients of thermal expansion.

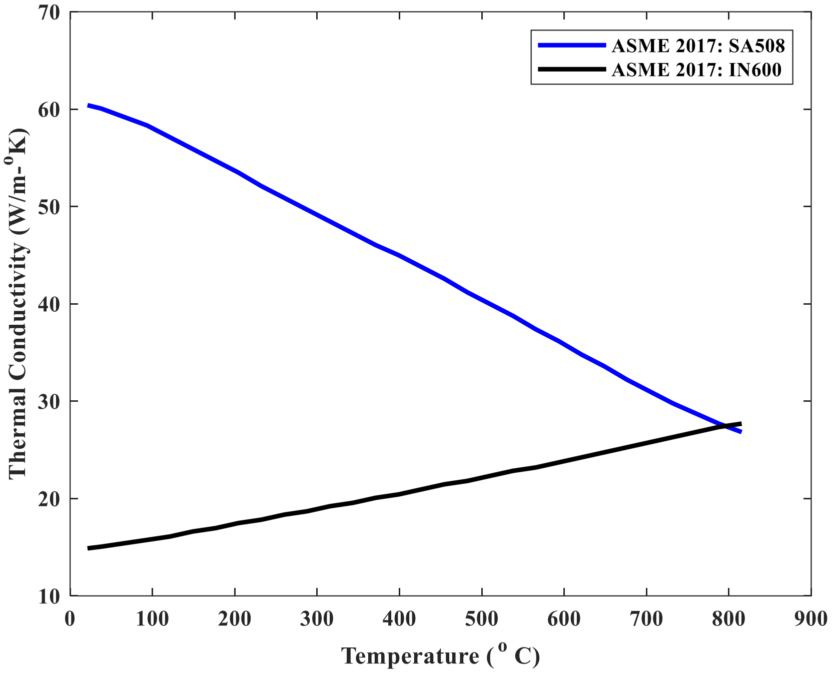


Figure 2. 4 Temperature-dependent thermal conductivity for low alloy steel (SA508) and Nickel alloy (IN 600).

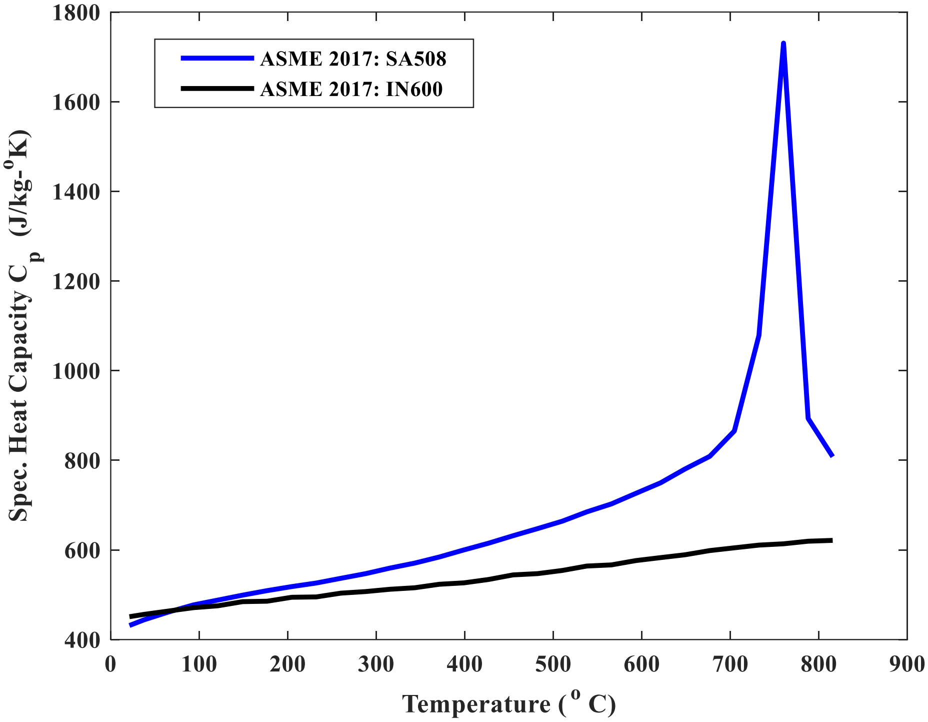


Figure 2. 5 Temperature-dependent specific heat capacity for low alloy steel (SA508) and Nickel alloy (IN 600).

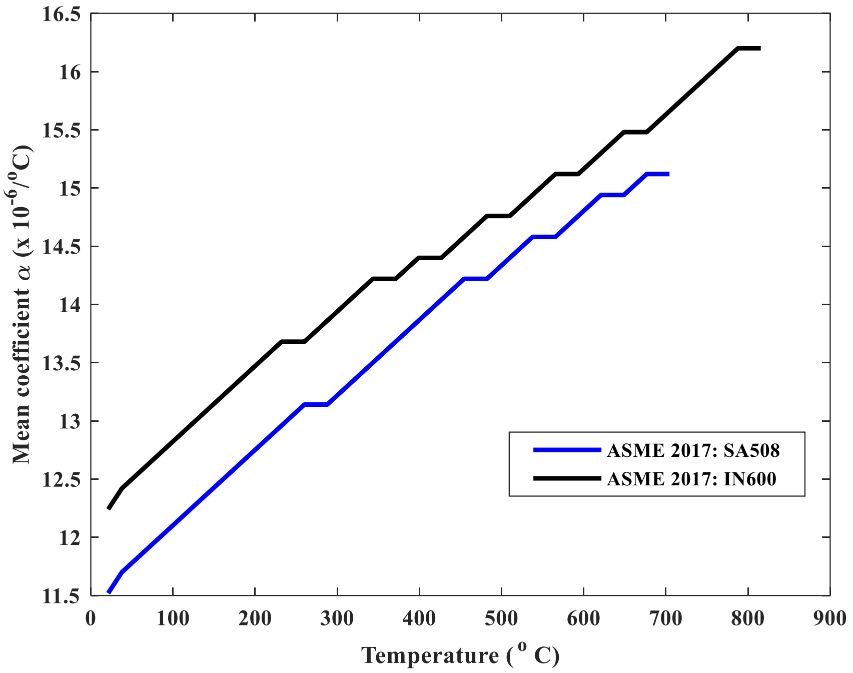


Figure 2. 6 Temperature-dependent mean coefficient of thermal expansion for low alloy steel (SA508) and Nickel alloy (IN 600).

2.1.3 FE Model Thermal Boundary Conditions

The heat transfer models were developed under two conditions such as under design-basis (DB) and grid-load-following (GLF) loading cycles. For the purpose we considered assumed input or process conditions. However, we anticipate the resulting DT model will be robust enough to predict given any random real time process measurements. The AI/ML based DT predictive model was trained using portion of the data FE simulated under DB loading cycle. Whereas the trained model was validated against FE data simulated under both DB and GLF loading cycles. The details of the assumed process inputs are as follows.

2.1.3.1 Boundary conditions for design-basis loading

Figures 2.7 to 2.9 show the DB loading-based various process temperature profiles, those were applied as inputs to the FE based heat-transfer-analysis model. Figure 2.10 to 2.12 show the corresponding primary and secondary loop pressure profiles.

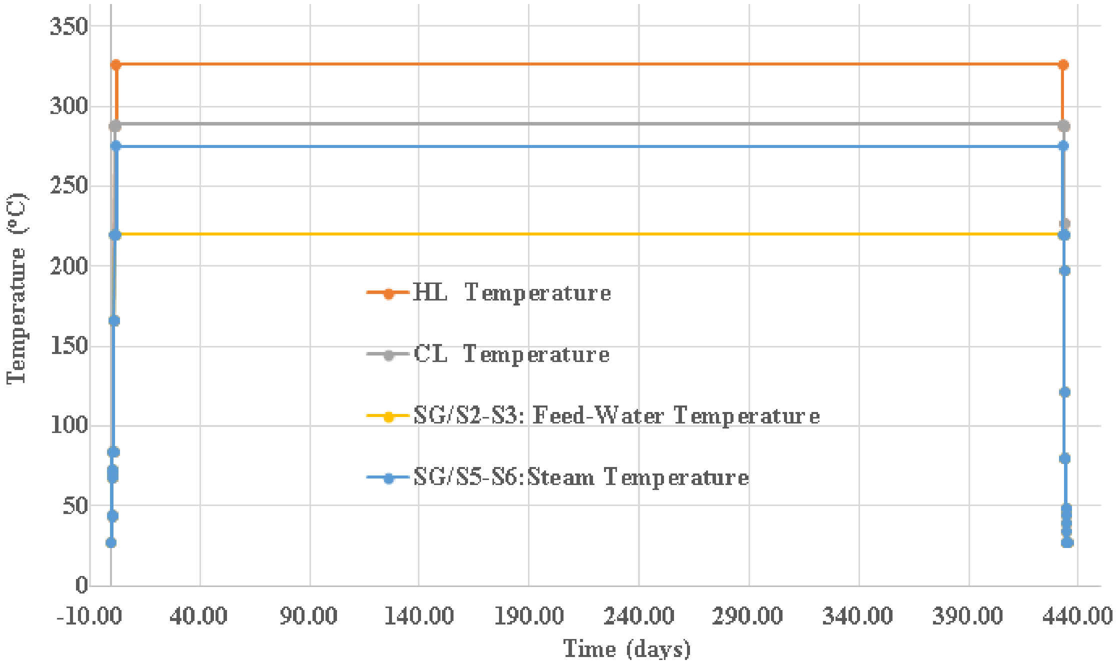


Figure 2. 7 DB loading cycle based various process temperature profiles; those were applied as inputs to the FE based heat-transfer-analysis model.

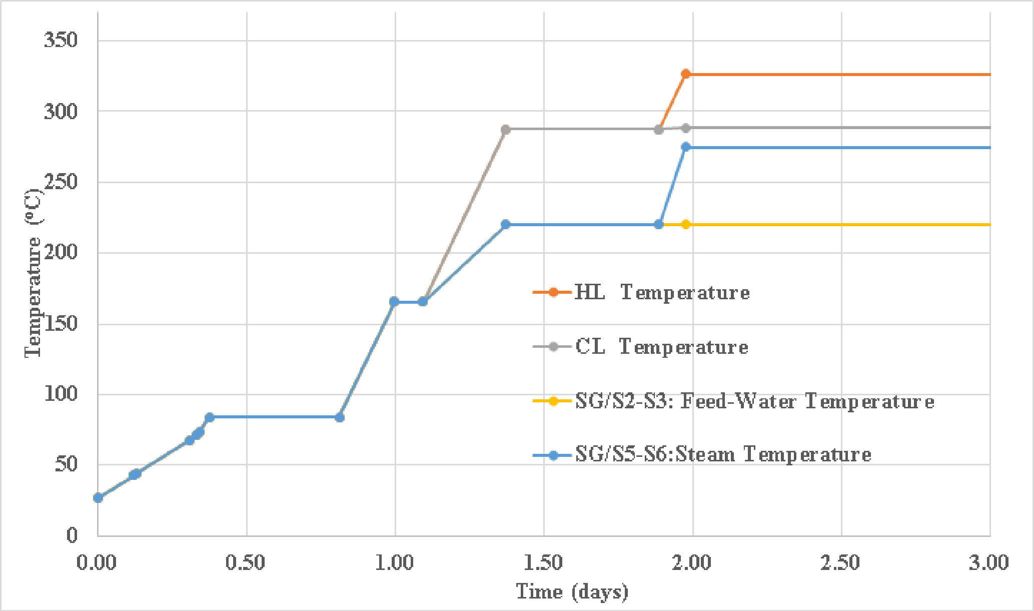


Figure 2. 8 Magnified view of Figure 2.7 showing the input/process temperatures during the heat-up operation.

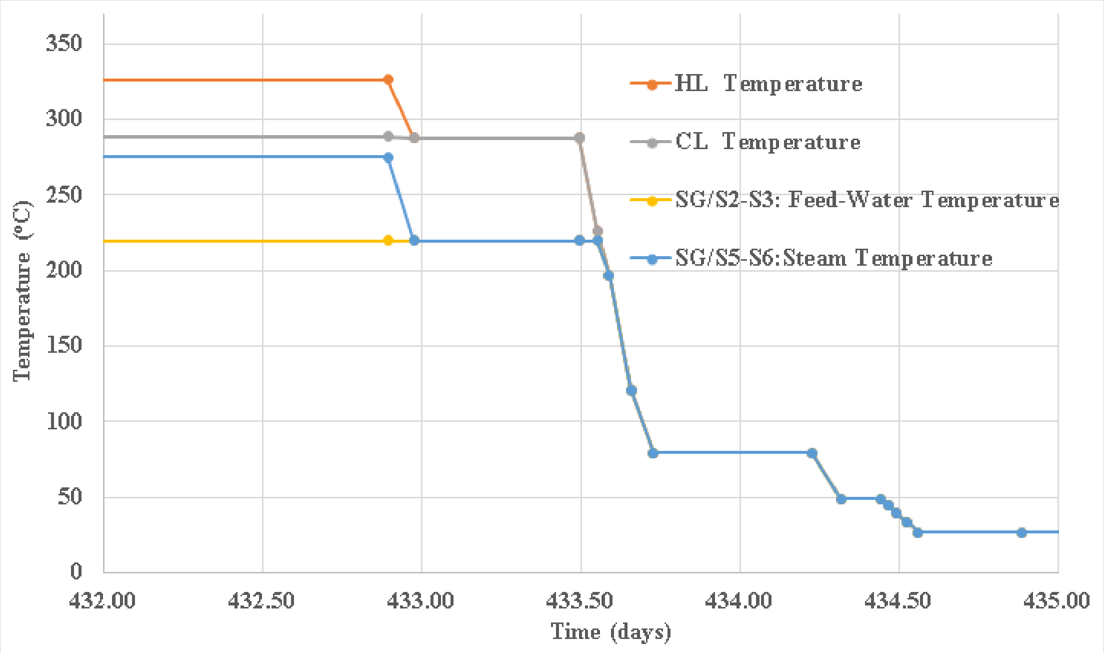


Figure 2. 9 Magnified view of Figure 2.7 showing the input/process temperatures during the cool-down operation.

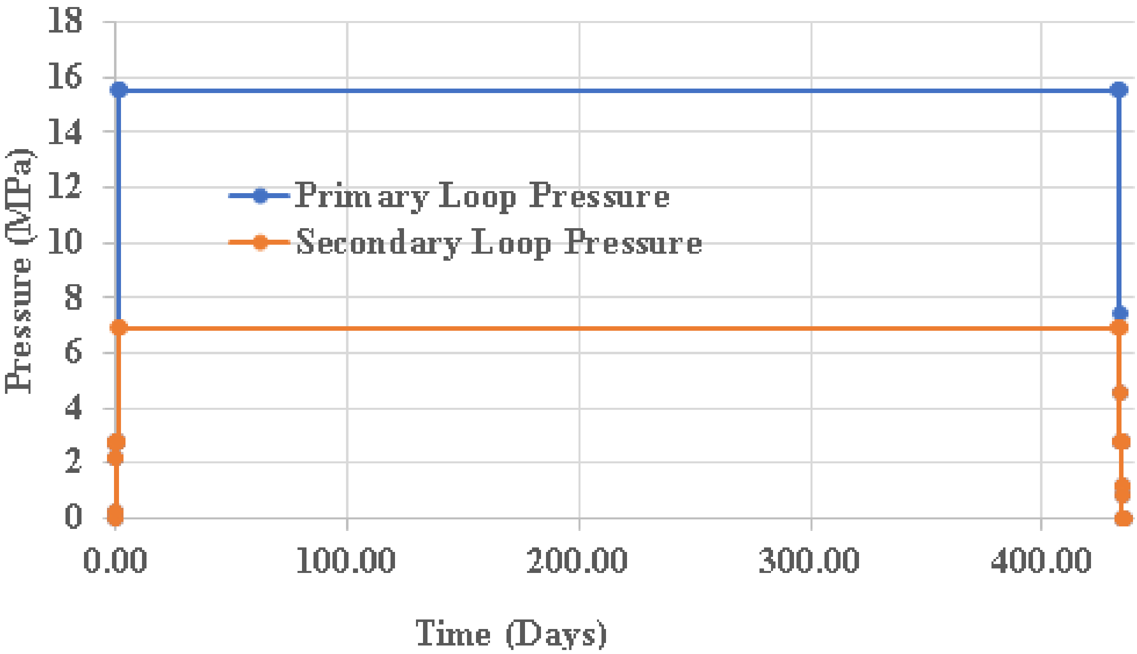


Figure 2. 10 DB loading cycle based primary and secondary loop pressure profiles.

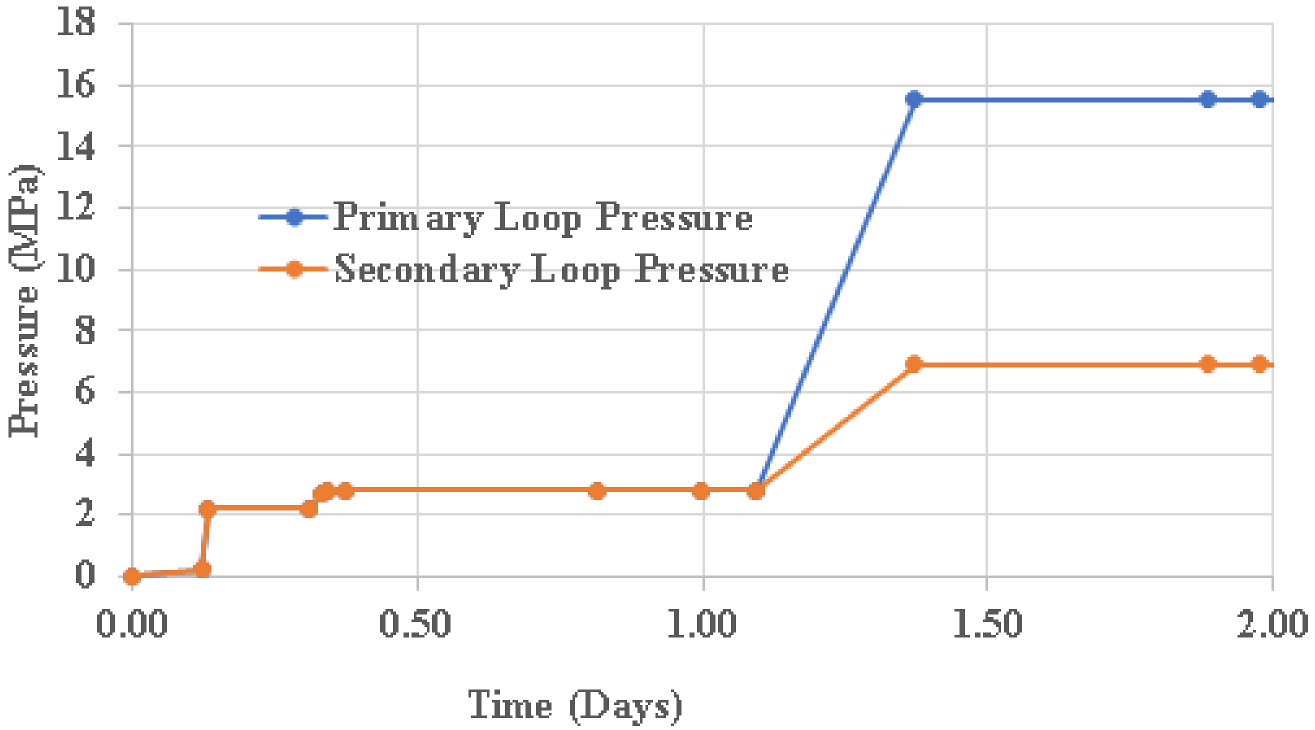


Figure 2. 11 Magnified view of Figure 2.10 showing the pressure profiles during the heat-up operation.

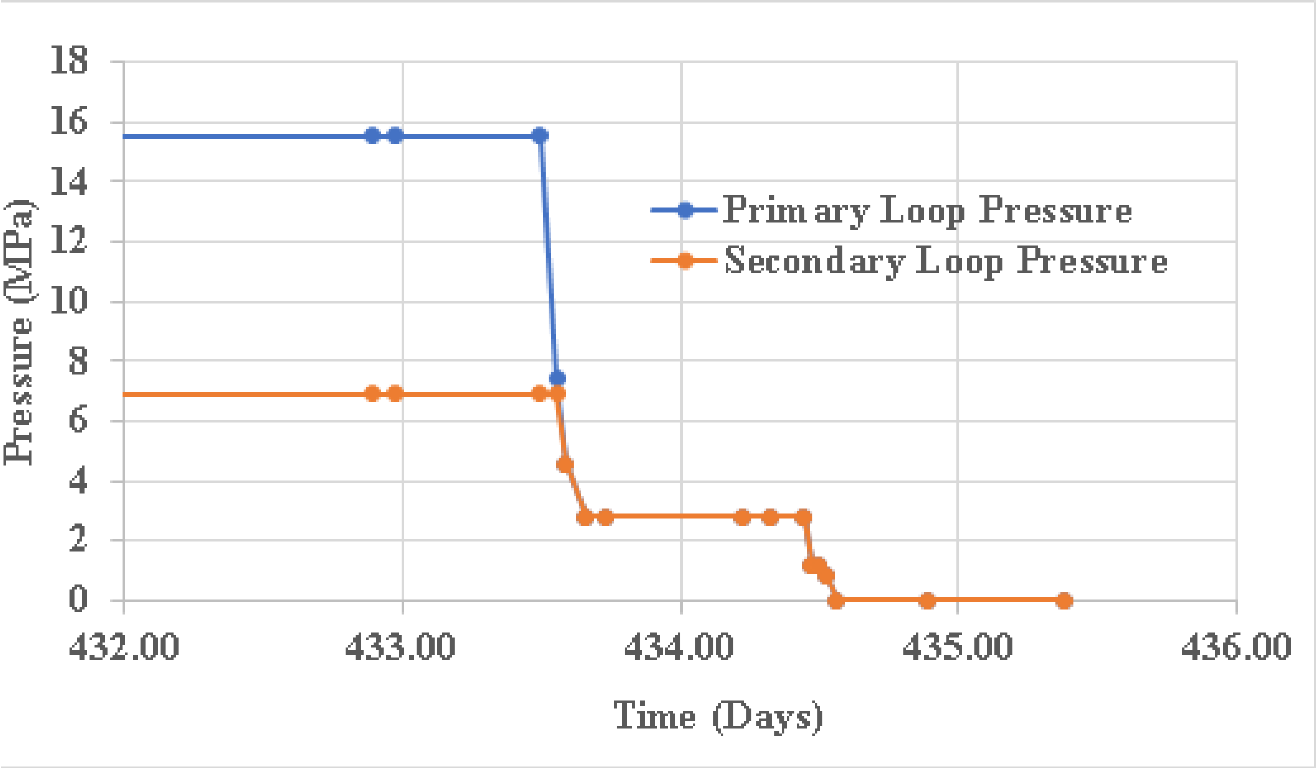


Figure 2. 12 Magnified view of Figure 2.10 showing the pressure profiles during the cool-down operation.

2.1.3.2 Boundary conditions for grid-load-following loading

In addition to DB loading cycle based transient heat transfer analysis we performed heat transfer analysis to simulate random spatial-temporal temperatures under GLF loading cycle. This is considering EDF (Électricité de France) based GLF power fluctuation profile [6]. Figure 2.13 shows the GLF power fluctuation profile (based on EDF data, that was used for generating the input wall temperature for HL, CL, and SG sections. Figure 2.14 shows the corresponding various process temperature profiles, those were applied as inputs to the FE based heat-transfer-analysis model.

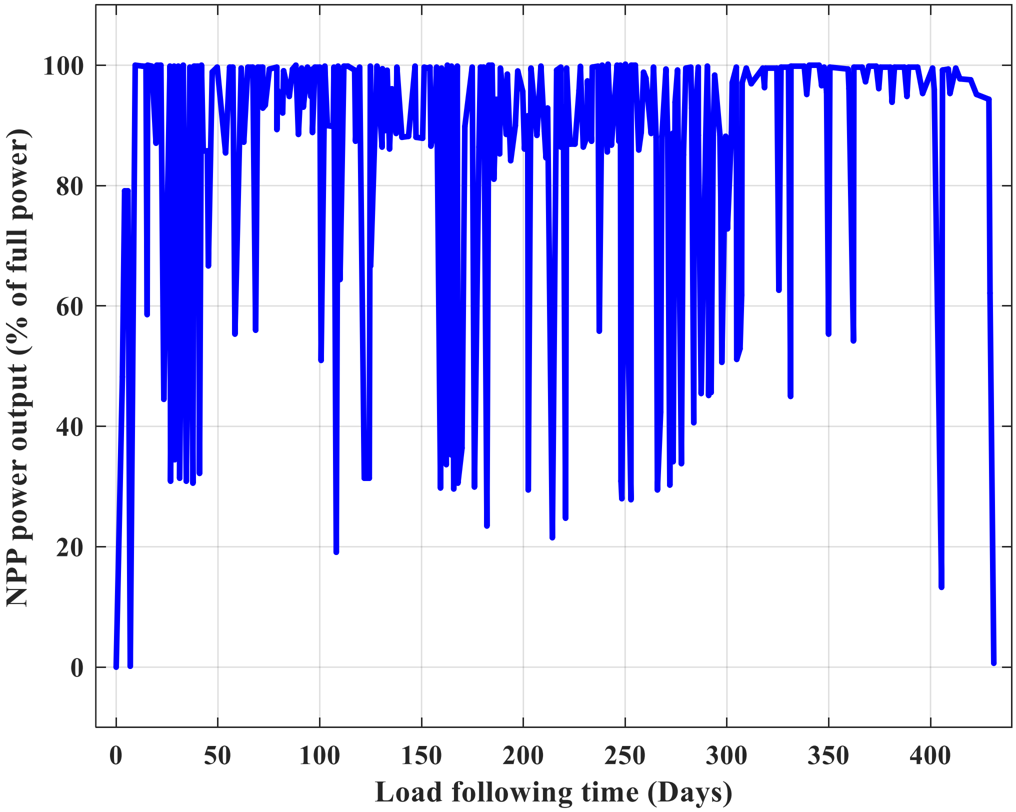


Figure 2. 13 Grid-load-Following power fluctuation profile (based on EDF data [6], that was used for generating the input wall temperature for HL and CL side of SG-bottom-head and other SG sections.

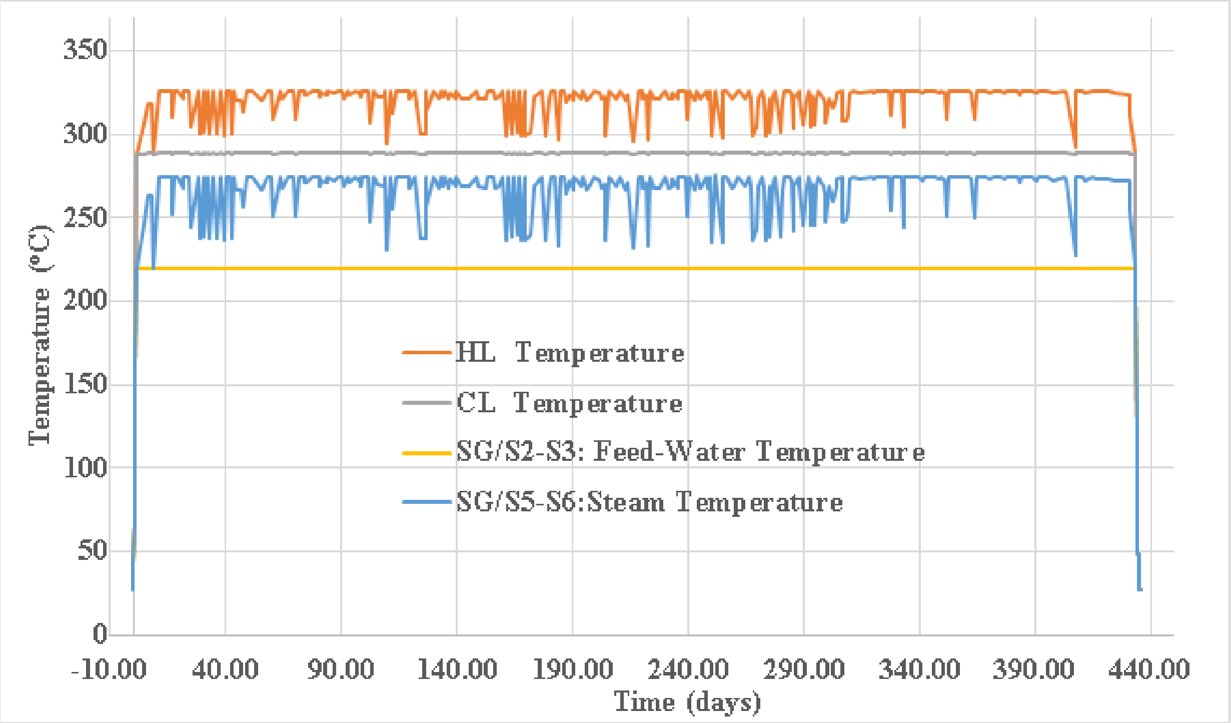


Figure 2. 14 Grid-load-following-based various process temperature profiles, those were applied as inputs to the FE based heat-transfer-analysis model.

2.2 Heat Transfer Analysis Results

Transient heat transfer analysis of the overall assembly (shown in Figure 2.1) was performed under both DB and GLF cycles. The results are as below.

2.2.1 Spatial and temporal temperature results under DB loading cycle

Figure 2.16 shows the FE simulated temperatures at all the 3720 nodes of the U-bend tube, simulated under DB loading cycle. The corresponding 3D nodal locations (or point-clouds) are highlighted in Figure 2.15. Figures 2.17 and 2.18 show the magnified view of Figure 2.16 showing the FE simulated nodal temperatures during the heat-up and cool-down operations, respectively. Figure 2.19 shows the spatial distribution of temperature (approximately at half cycle or at 217.7 days) at the OD and ID surfaces of SG. Whereas Figure 2.20 shows the FE simulated spatial distribution of temperature (approximately at half cycle or at 217.7 days) at the OD surface of SG tube along its entire length. The resulting temperature data (at 3720 FE nodes or 3D point-clouds) were used later training and/or validating the DT based temperature prediction models (refer sections 4 to 7).

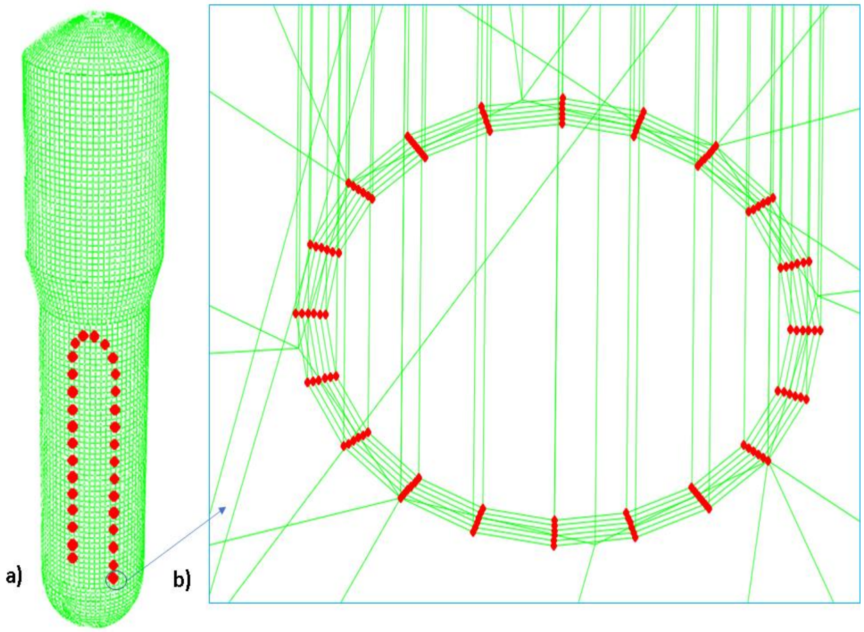


Figure 2. 15 Highlighted 3720 FE nodes (or 3D point-clouds), which temperature results are presented a) all the nodes of a SG U-bent tube, b) cross-sectional nodes of the U-bend tube.

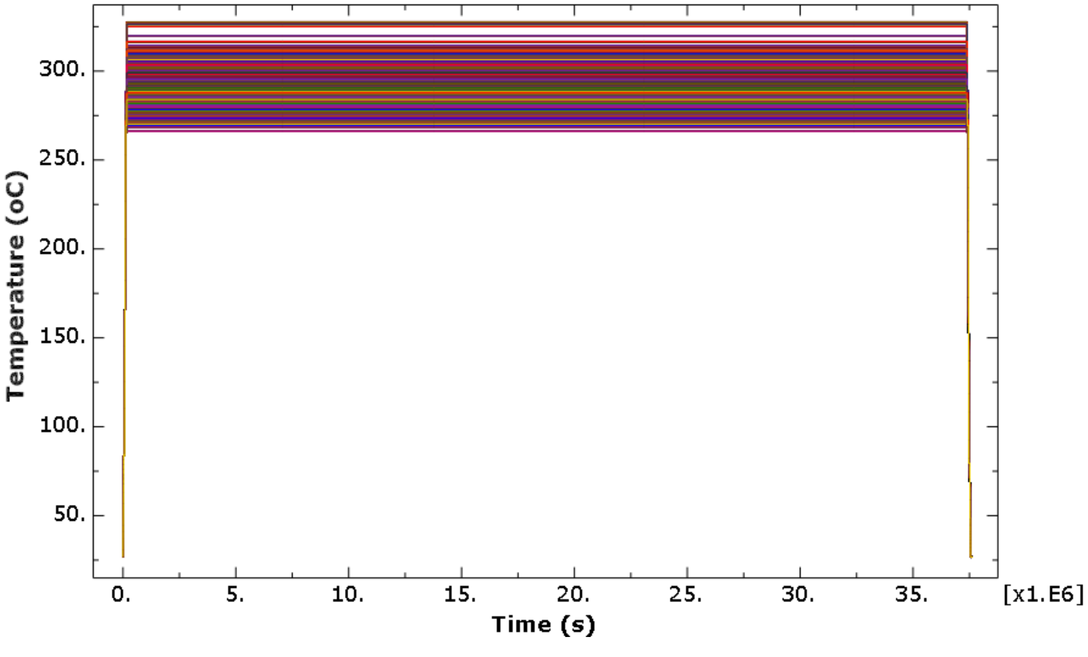


Figure 2. 16 Design-basis-loading FE simulated temperatures at all the 3720 nodes highlighted in Figure 2.15.

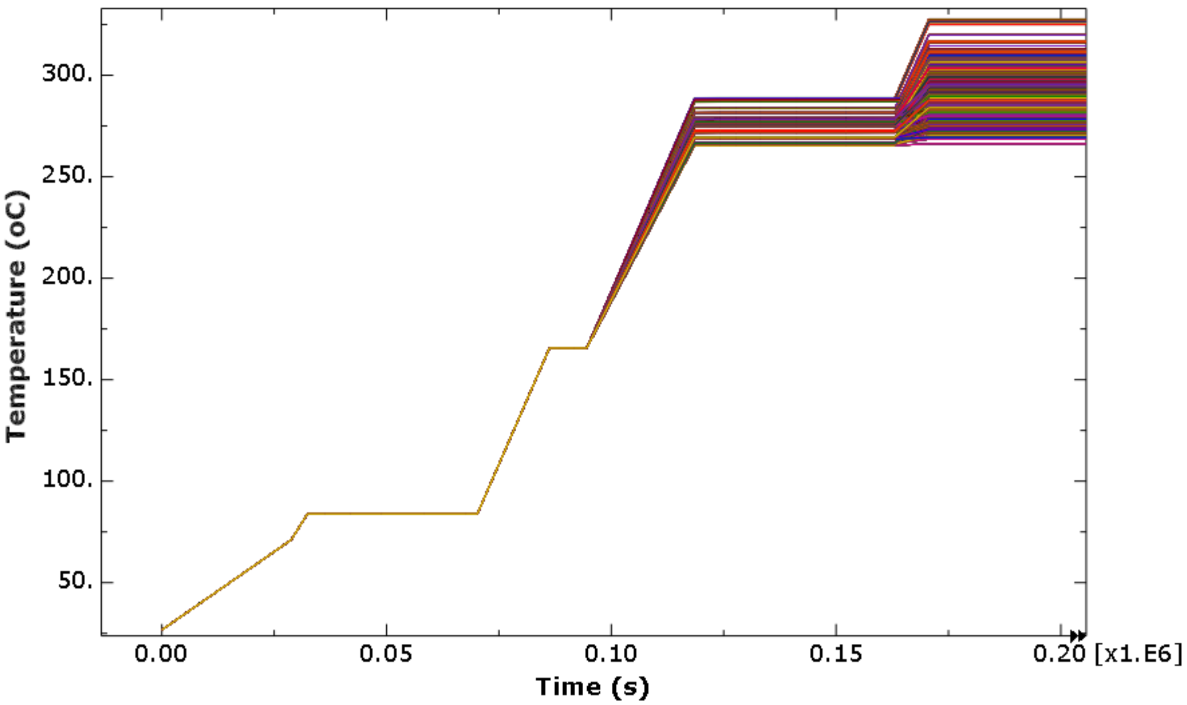


Figure 2. 17 Magnified view of Figure 2.16 showing the FE simulated nodal temperatures during the heat-up operation.

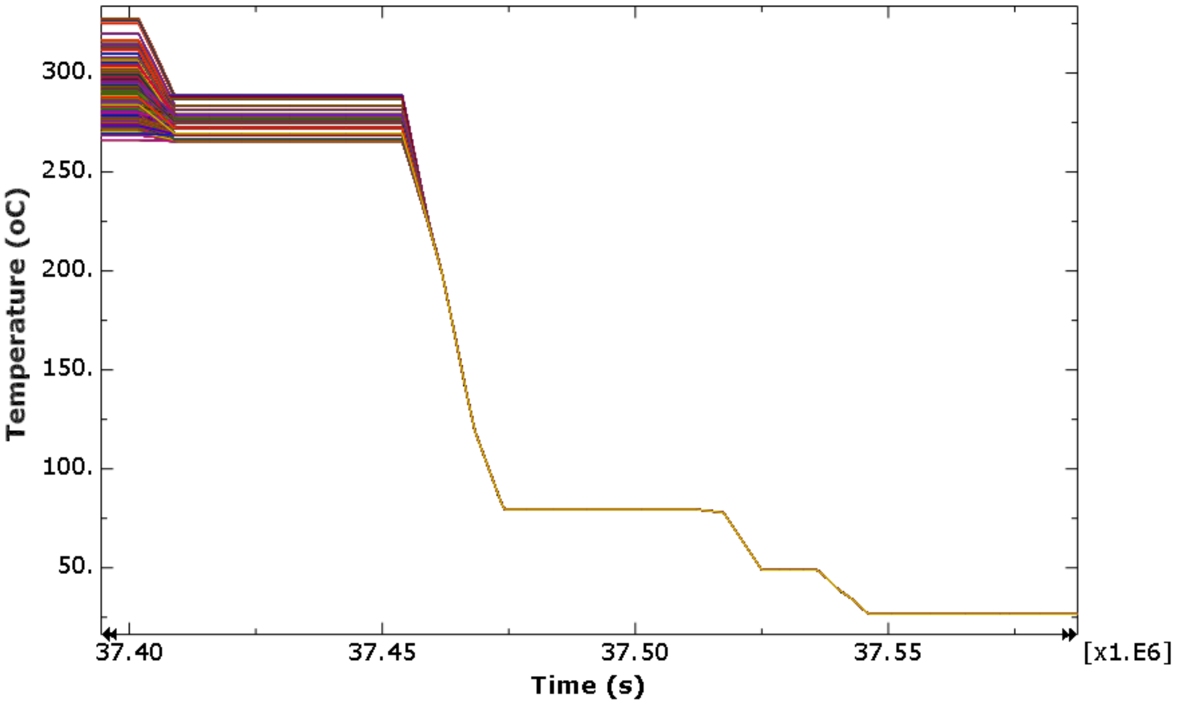


Figure 2. 18 Magnified view of Figure 2.16 showing the FE simulated nodal temperatures during the cool-down operation.

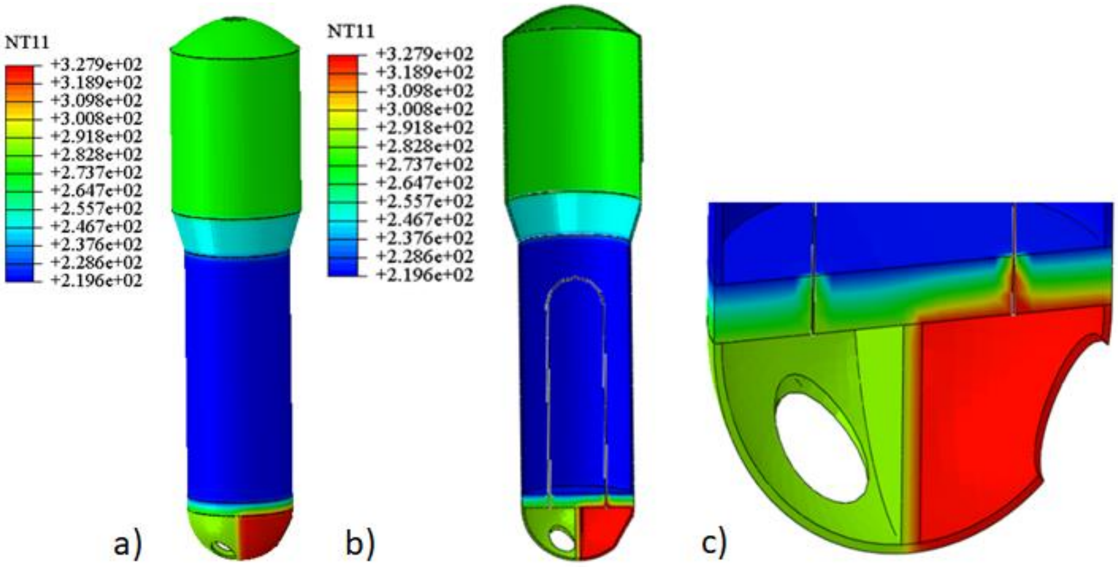


Figure 2. 19 Design-basis-loading FE simulated spatial distribution of temperature (approximately at half cycle or at 217.7 days) at a) OD surface of SG, b) ID surface of SG, and c) magnified view at SG bottom-head and tube-sheet.

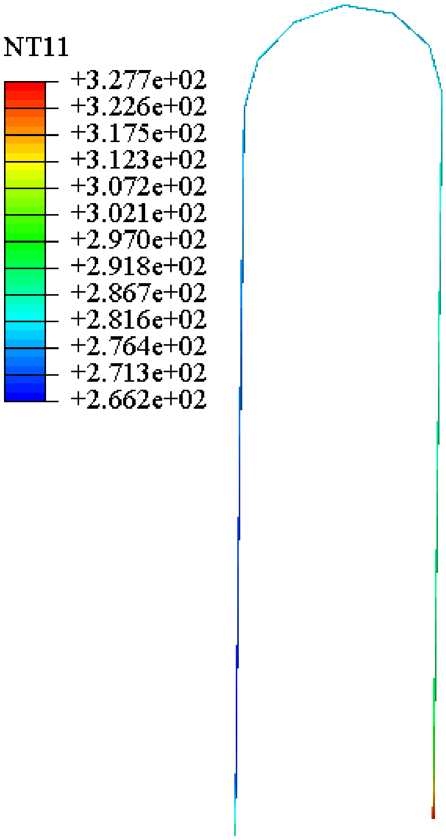


Figure 2. 20 Design-basis-loading FE simulated spatial distribution of temperature (approximately at half cycle or at 217.7 days) at the OD surface of SG tube along its entire length.

2.2.2 Spatial and temporal temperature results under GLF loading cycle

Figure 2.21 shows the FE simulated temperatures at all the 3720 nodes of the U-bend tube, simulated under GLF loading cycle. Figure 2.22 shows the magnified view of Figure 2.21 showing the temperature variation during full-power operation. Figure 2.23 shows the example FE simulated spatial distribution of temperature (approximately at half cycle or at 217.7 days) at the OD surface of SG tube along its entire length. The temperature variations along the length of SG tube can generate substantial bending stress and associated bowing that can lead to touching of the SG tube with intermittent tube supports. This in turn can lead to wear related damage and associated crack initiation hotspots. In top of this, high-cycle fatigue associated with flow induced vibration can have coupled effect of combined low and high cycle fatigue damage. Our anticipation is that using an appropriate AI/ML based DT model, the shown temperature fluctuation and the associated stress-strain states of a tube can be inferred in real time given the underlying input variables or process measurements. To note only a single or onetime FE simulation is required for only initial training of the AI/ML based DT model.

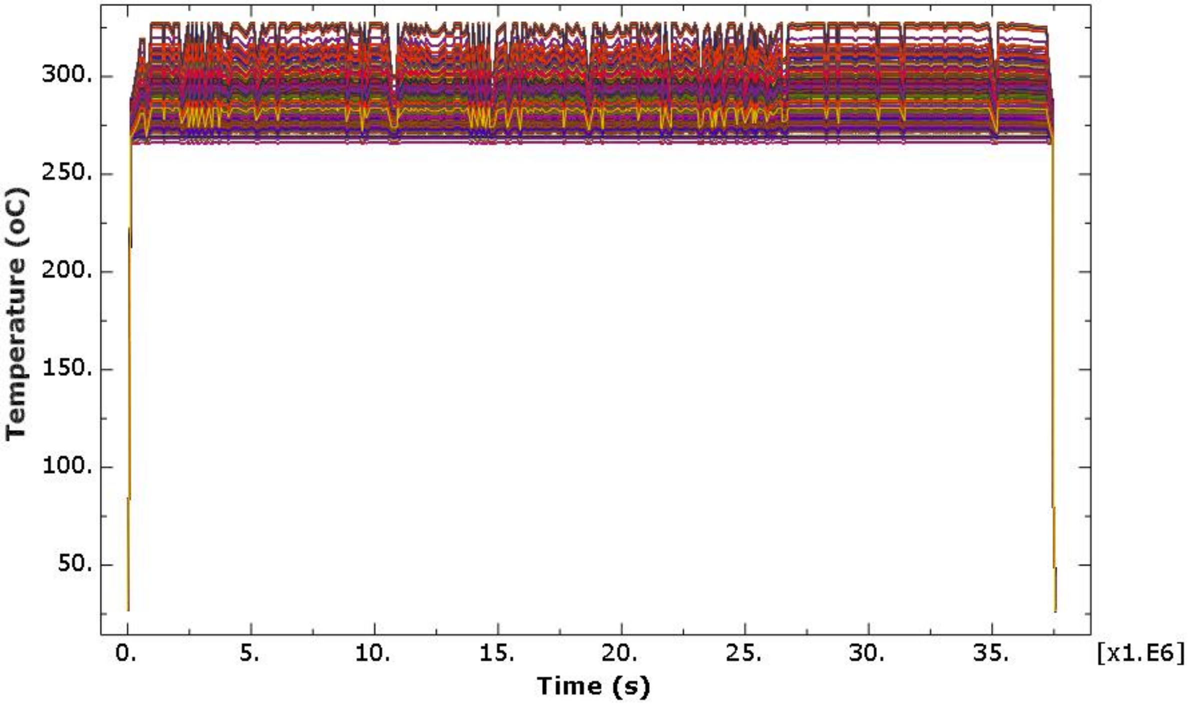


Figure 2. 21 Grid-load-following-loading FE simulated temperatures at all the nodes highlighted in Figure 2.15.

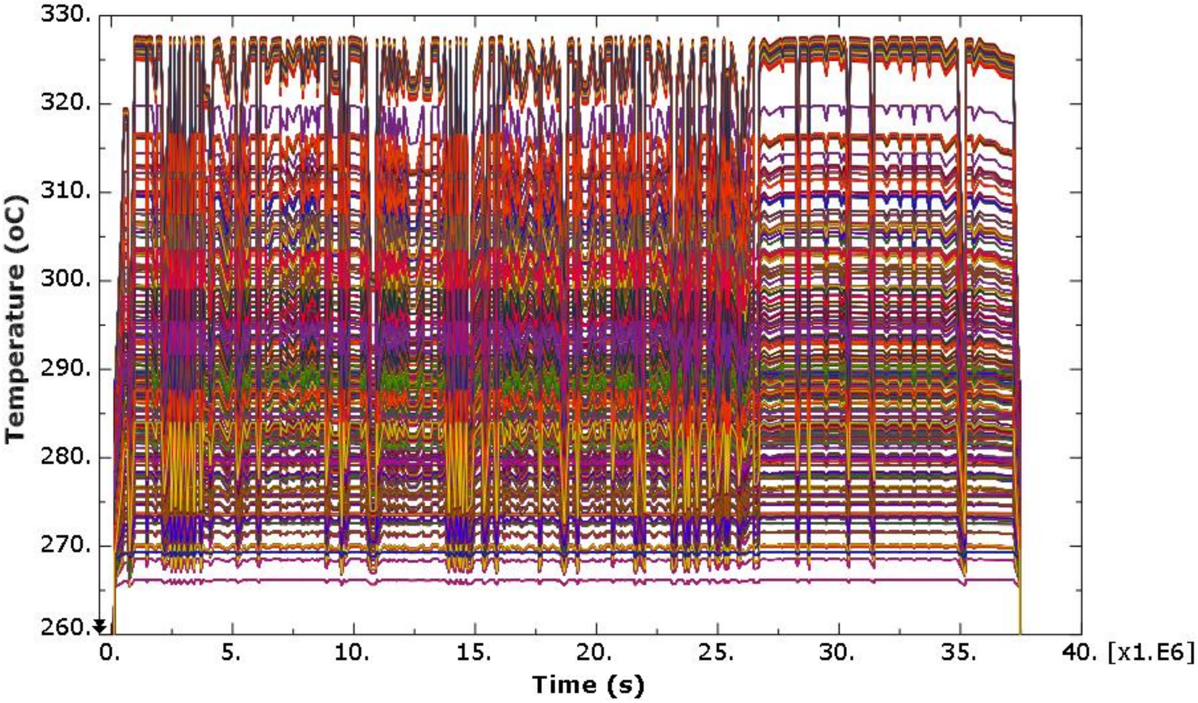


Figure 2. 22 Magnified view of Figure 2.21 showing the temperature variation during full-power operation.

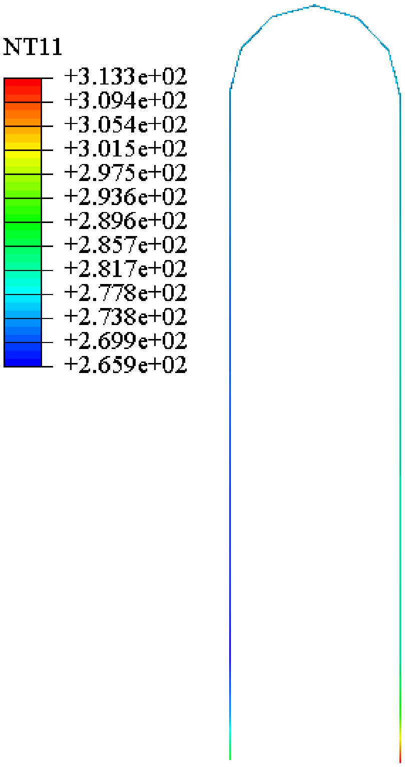


Figure 2. 23 Grid-load-following-loading FE simulated spatial distribution of temperature (approximately at half cycle or at 217.7 days) at the OD surface of SG tube along its entire length. at half cycle OR at 217.7 days) at the OD surface of SG tube along its entire length.

3 MongoDB Based NoSQL Database Implementation

A DT system has to be primarily based on AI-ML data-driven analytics framework that can indirectly use data from physics based virtual sensor measurements (from multi-physics digital model of the actual component) and real sensor measurements. The sensor measurements are either continuous measurements from an online monitoring (OLM) system and/or non-continuous measurements from offline in-service inspections that employ Non-Destructive Evaluation (NDE) methods. However, the overdependence of a DT system on data (from virtual and/or real sensor measurements) necessitates access to a suitable database system in addition to the development of core analytics techniques and generation of underlying data. The database must store hundreds to thousands of sensor measurements (either from virtual or real sensor measurements), with structured/un-structured data source, with different time scales (some measurements in the scale of milliseconds, some in the scale of hours to days), with continuous/non-continuous data source and at the same time work in a real time and agile environment. This type of multi-faceted requirements needs a suitable storage solution or database that must be efficient in providing big data storage, high scalability, high concurrent reading with low latency and high availability. Although traditional relational database management systems (RDBMS) such as Structured Query Language (SQL) based databases are heavily used in industry (e.g. banking and ecommerce industries), the trend is increasingly shifting toward “NoSQL database”. NoSQL database typically refers to any non-relational database with “NoSQL” representing “non-SQL” or “not only SQL”. The increasing trend toward using NoSQL database arises from the need to circumvent some of the issues associated with RDBMS type database. In this section, a summary of major differences between SQL and NoSQL type database and the reason for choosing a NoSQL database for nuclear reactor DT applications are discussed. Also, some of the example results related to implementation of NoSQL based database such as using commercially available MongoDB database are discussed.

3.1 Review of SQL Versus NoSQL Database

The Structured Query Language (SQL) based RDBMS were developed in 1970s and since been highly popular [7]. Since the SQL language is primarily used for querying and maintaining the RDBMS type database, the RDBMS type database is also popularly known as SQL database. The RDBMS type database is a general-purpose database, in which all the data are stored in the form of tables with fixed rows and columns. This type of database facilitates primary key, which helps in unique identification of the rows. In RDBMS the inter-related tabular data are tied with a rigid schema (data organization blueprint or rules). This helps with reducing duplication, and on the other hand helps with reducing the hardware (e.g. storage) requirements. In addition, most of the RDBMS type database follows ACID (Atomicity, Consistency, Isolation, and Durability) properties, which ensures data validity despite errors, power failures, and other mishaps [8, 9]. All the above-mentioned good properties have made RDBMS type database such as Oracle [10], MySQL [11], Microsoft SQL Servers [12], etc. immensely popular. However, recent exponential growth of data (both structured and non-structured) based applications such as those related to social network, e-commerce, and Internet of Things (IOT) applications, face greater issues associated with traditional RDBMS type database [13, 14]. Some of the major issues are:

a) Constraints of horizontal scalability over several servers associated with the ever increasing need to store large amount of data. A large amount of data can force traditional RDBMS type databases to scale vertically, thus requiring migration to a larger, more expensive server or supercomputer (when the storage capacity of an existing RDBMS server is exceeded). For example, IOT and DT type applications need to be heavily dependent on multiple data streams, both from point sensors (e.g. thermocouple, accelerometers, etc.) and from spatial sensors (e.g. 3D point cloud data from camera, LiDAR, etc.) along with data from virtual sensors (e.g., spatial-temporal data from thousands to millions of 3D nodal points or point-clouds). The data from these sensor sources can be both continuous (taking measurements in real time) and discrete (e.g., periodic NDE measurements, image/video capture through drone, etc.). The resulting large amount of data can quickly overwhelm an existing server and force augmentation of the storage capability either vertically and/or horizontally. However, RDBMS type server has the constraint of horizontal scalability and may only grow vertically, which can be expensive.

b) Data querying in RDBMS type database can be time consuming when there is a complex relation between constituent inter-related tables and when the table size grows over time. Since SQL based database requires joining data from multiple inter-related tables, as the relational complexity between tables and the size of tables grows the relational joining operations and the associated query can become computationally expensive (both in terms of computational time and programming efforts). These issues could further become a challenge particularly in dealing with unstructured data and heterogeneous data types with complex dependency.

c) Rigid schema or rule-based RDBMS makes it difficult to change in later stage when the requirements for database changes. This will be a challenge if an application requires adding new features that depend on new attributes that do not exist in the current database.

d) The fixed row and column size in RDBMS type database can take unnecessary storage space (e.g. null space) even if there is no data in a particular table location.

The above-mentioned drawbacks in SQL or RDBMS type database led to the development of NoSQL type database. The “NoSQL” synonym refers to “Non-SQL” or “not only SQL”. The development of NoSQL type database primarily started in the late 2000s. The main focus was horizontal scaling (scaling up server requirements, across cheap commodity servers rather than being forced to vertically add server capability such as adding high-cost supercomputers), fast queries, flexible schema (that can allow for frequent application changes and intermediate scale up associated with application requirements). Currently there are different types of NoSQL database platforms available. Some of the popular types are:

i. ***Key-value database:*** A key-value database stores items as alphanumeric identifier or keys and the associated values in a data structure format commonly known as a dictionary or hash table. A key-value pair can contain a collection of records, which in turn can have many different fields within them and the associated data. Unlike the RDBMS type database, a key-value type database doesn’t require a predefined data structure containing fields with well-defined data types. Rather, a key-value database treats the data as a single opaque collection, which can have different fields and data types. This feature allows for substantial flexibility (over RDBMS type database) such as adding new datatypes and the associated attributes in a later stage without completely redeveloping the database from scratch. An example of key-value database is Amazon's Dynamo database [15, 16].

ii. ***Wide-column database:*** A wide-column database uses tables, rows, and columns, similar to RDBMS type database, but unlike RDBMS, wide-column databases are highly flexible. They have no predefined keys or column names. Similar to key-value database, the schema-free features of wide-column database allows modification of column names and adding new columns in same table, even in real-time. A wide-column database is a type of two-dimensional key–value database that allows multiple attributes per key. An example of wide-column database is Google's Bigtable [16, 17].

iii. ***Graph database:*** Graph databases are a type of NoSQL database, which replace relational tables with structured relational graphs. The graph depicts an object-oriented network of objects (represented by nodes of a mathematical graph), their relation to each other (represented by edges of a mathematical graph/grid) and the properties of individual objects (which are the object attributes expressed as key-value pairs). Graph databases are primarily useful for visualizing the relation between different data objects (e.g., fraud pattern detection among credit card identity theft of a bank). An example of graph database is Neo4j [18, 19].

iv. ***Document database:*** Broadly, a document type database stores document-oriented information using data exchange format such as XML Extensible Markup Language), JSON (JavaScript Option Notation) or BSON (Binary JSON). Document-type databases are a subclass of the key-value type database. However, unlike simple key-value database in which the data are inherently opaque to the database, the document-type database relies on internal structure in the document to extract metadata (i.e., the data about data or the information about the data) that the database engine uses for further optimizing the storing and querying process. This is for example storing and querying information with multiple fields as a single metadata array with multiple keys and values. Also, unlike key-value database, the value column in document databases can contain semi-structured data with a single column containing hundreds of attributes. In addition, the number and type of attributes can vary from row to row. This type of features allows high level of flexibility (compared to traditional RDBMS type database) for adding new data attributes. Particularly, being a schema-less database, making it easier to scale as required. An example of document-type database is MongoDB [20, 21] and CouchDB [22, 23] based databases.

Although, the above-mentioned NoSQL type databases are very similar to each other with the primary aim of alleviating the issues associated with SQL type databases, the document type MongoDB is one of the most popular databases [13, 24, 25] among the available NoSQL databases. This is possibly due to a mature ecosystem of tools, a vibrant community of users and the availability of open-source information on how to use the MongoDB. This made us to consider MongoDB as a database platform for developing the DT framework.

3.2 The Big Picture – Data Flow and Database Requirements for Digital Twin Applications

3.2.1 Big Picture

The size and scaling of a database depend on the application and the number of hierarchical levels of data flow. For example, a DT based predictive maintenance system can have a multi-level-framework, starting with the lowest level focused on small regions of a single component to a national level center for monitoring and predicting the structural health of all the components in all the plants (in US) from a centralized location. Figure 3.1 depicts the schematic of a hierarchical system [26]. Each plant can be divided into subsystems containing components with individual sensor nodes. The sensor nodes can connect to a subsystem network via wired or wireless connection. The subsystem network would relay all of the information from all the nodes to the plant control center. Plant managers can be presented with real-time continuous updates regarding the overall plant structural integrity (SI) and can easily and quickly diagnose problems down to the smallest component. Additionally, the health information for each plant can be transmitted to a regional control center and national control center for further monitoring and regulation. The national monitoring system would allow in depth aging and health analysis of the nation’s fleet of nuclear plants efficiently and in real time.

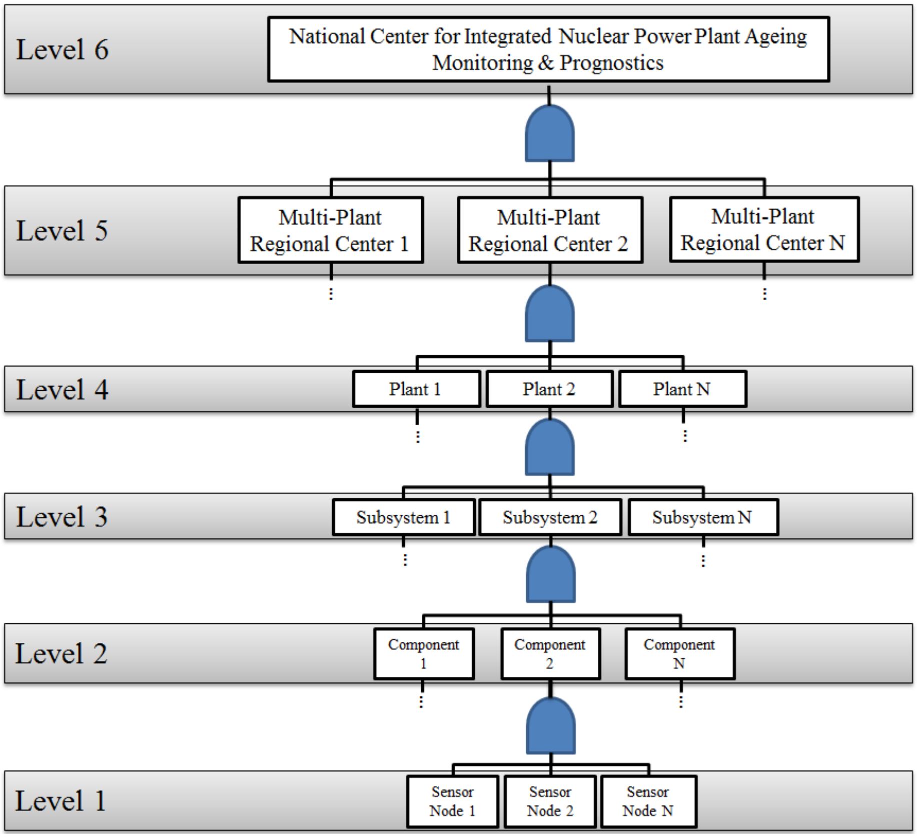

Level 6
National Center for Integrated Nuclear Power Plant Ageing Monitoring & Prognostics
Level 5
Multi-Plant Regional Center 1
Multi-Plant Regional Center 2
Multi-Plant Regional Center N
Level 4
Plant 1
Plant 2
Plant N
Level 3
Subsystem 1
Subsystem 2
Subsystem N
Level 2
Component 1
Component 2
Component N
Level 1
Sensor Node 1
Sensor Node 2
Sensor Node N

Figure 3. 1 A fault tree diagram of a national level DT based online health monitoring and prediction system [26].

At the component level, the time-series state of the structure at a given time can be estimated based on sensor measurements pertinent to the component of concern. The sensor measurements can either continuous, from an online monitoring (OLM) system, or non-continuous, from offline in-service inspections that employ Non-Destructive Testing (NDT) methods. In addition, for estimating the state of a structure at a given time, the time-series future state of the structure can be predicted at any given time. However, unlike the current state estimator, which has to depend on OLM and/or NDT based real sensor measurements, future state prediction requires having additional information such as process measurements (both historical, current and future/anticipated), virtual sensor measurements such as spatial-temporal structural states (e.g., inside/outside surface temperature, strain, stress, etc.) of the components of interest under simulated thermal-mechanical loading conditions. Additionally, the predictive leg of the DT model can incorporate real time material condition (e.g., through material dependent stress-strain hardening/softening), environment (e.g., light water reactor water chemistry, temperature, etc.) and loading condition (e.g., strain/load amplitude and rate, loading sequence, etc.) to further improve the accuracy of the prediction. The predictive model needs also to predict the probabilistic states associated with material variability and other uncertain conditions. Based on the predicted states and with knowledge about the critical failure criteria the remaining life of a component can be predicted at a given time [26, 27]. Figure 3.2 depicts a schematic showing the forecasted time-series structural states based on the information from the structural state at any given time [26].

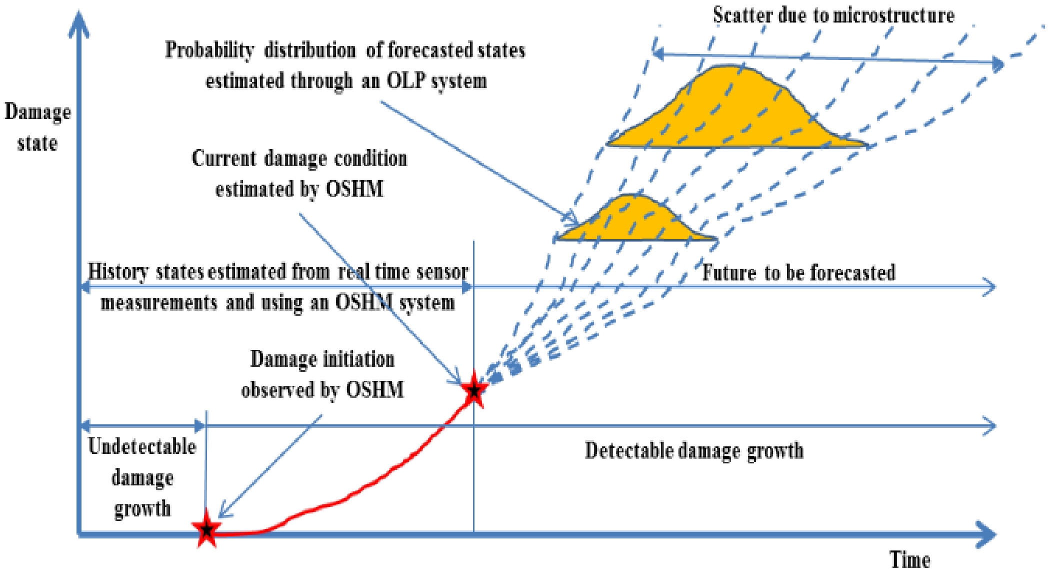

Scatter due to microstructure
Probability distribution of forecasted states estimated through an OLP system
Damage state
Current damage condition estimated by OSHM
History states estimated from real time sensor measurements and using an OSHM system
Future to be forecasted
Damage initiation observed by OSHM
Undetectable damage growth
Detectable damage growth
Time

Figure 3. 2 Schematic of already degraded states of structure estimated through OLM and/or NDT measurements and forecasted states [26].

3.2.2 Heterogenous Source and Timescale of Data

The DT framework depends on heterogeneity of the source and timescale of the data that is either generated virtually or by real sensors. For example, at present we are developing a DT framework for structural state prediction of a steam generator (SG) of a pressurized water reactor (PWR). To predict the structural state (such as stress and crack size) one needs to know the temperature at any given location and time (spatial-temporal distribution). This includes information about the variation in temperature over the length, width, and thickness of a component. It is rather impossible to place sensors at all the locations of interest along a component. However, if historical (already existing) process measurements (which can be considered as input variables), and the corresponding length, width, and thickness variation of temperature data (which can be considered as output variables) are available, then an AI-ML based input-out mapping function can be established. This mapping function can then be used for predicting the variation in temperature along the length, width and thickness of a component using the measured process parameters in real time.

The prediction of structural states (e.g. crack size, ratcheting stress or strain) at a future time depends on the past state information (on what had already happened and their progression rates) and on the state information at present. The actual state of the structure at a given time) can be estimated from the actual NDE and/or OLM sensor measurements taken at that time. A DT system can be based on both the historical and the current NDE and OLM data. The historical NDE and OLM measurements can include data from both laboratory tests (e.g. laboratory tests of a similar component under similar conditions as the actual component) and/or direct measurements from the actual component in the field. As noted previously, the sensor measurements can be either continuous measurements from an online monitoring (OLM) system or non-continuous measurements from offline in-service inspections that employ NDE methods. These NDE and OLM data can be used for training the data analytics-based state estimator model. As an example, Figure 3.3 shows laboratory NDE data associated with EC measurements on a SG tube section with a shallow volumetric flaw (tube wear). The EC signal from the wear scar is located in the center of the data segment. Figure 3.4 shows the EC data acquired on the same SG tube after a SCC was produced near the wear scar. The signal from the relatively deep SCC has a significantly lower amplitude than that of the nearby wear mark. Detection of such cracks poses a major challenge to NDE techniques when located in close proximity to a volumetric flaw.

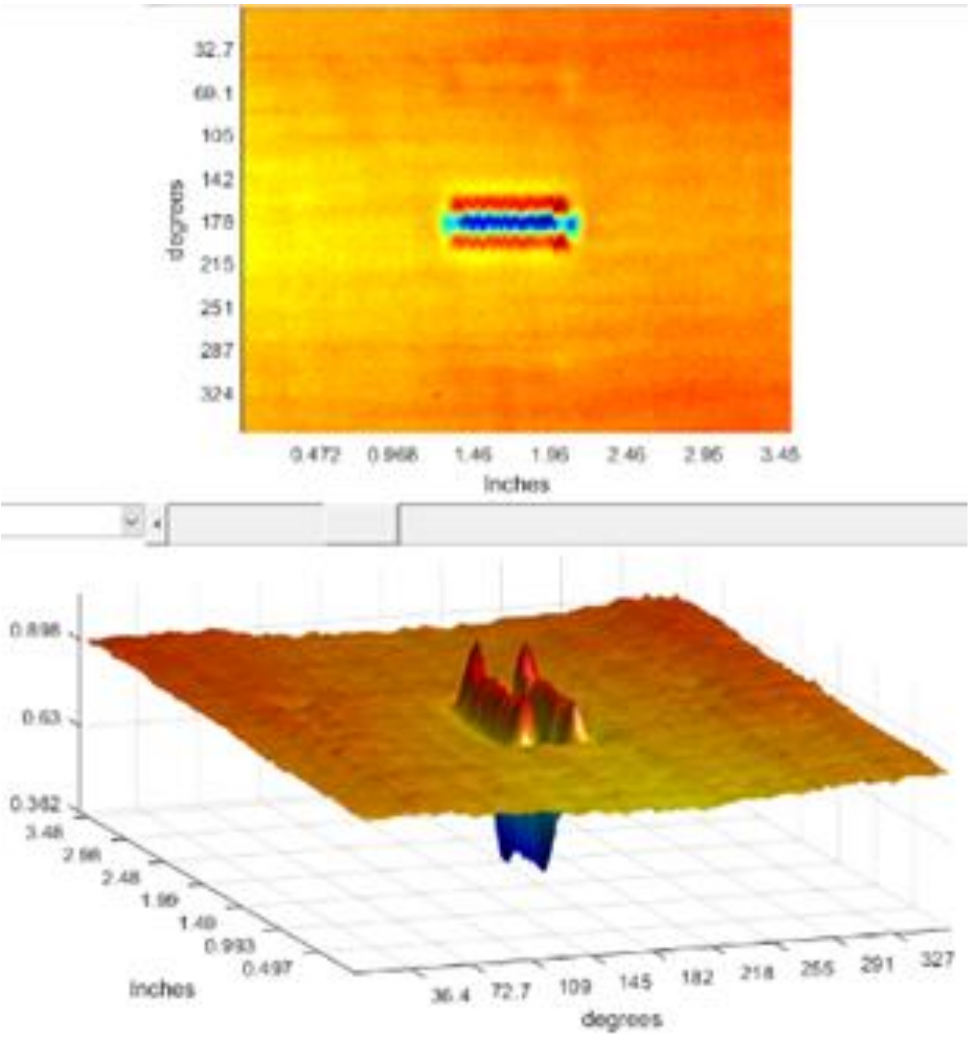


Figure 3. 3 Example offline NDE based EC measurement results for a SG tube before being subjected to SCC test. The EC data segment encompasses the signal from a wear mark.

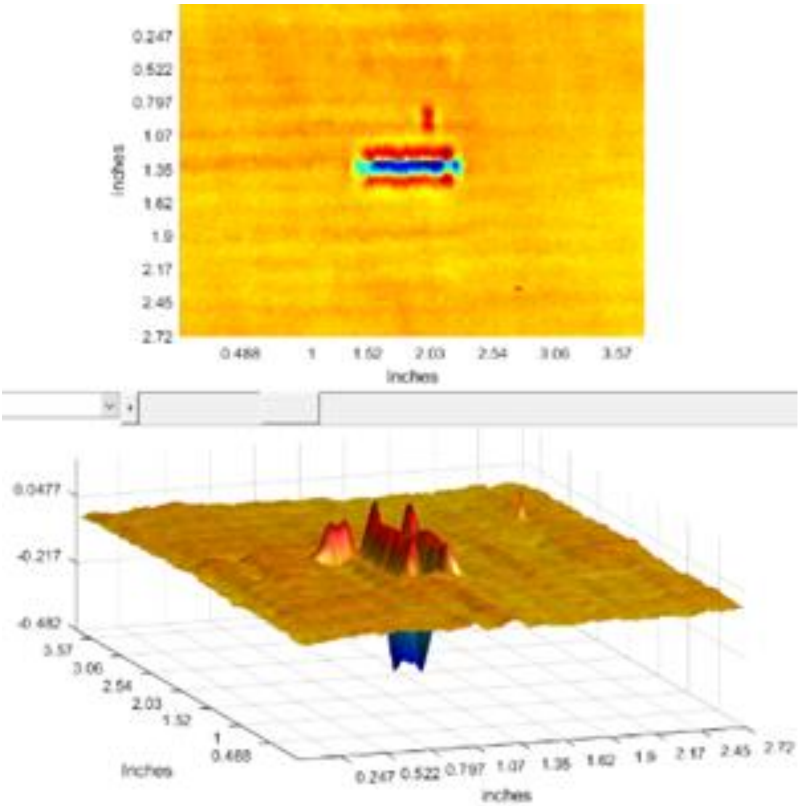


Figure 3. 4 Example NDE based EC results for SG tube after being subjected to SCC test. The EC data segment encompasses the signal from SCC damage near the wear mark shown in Figure 3.3.

As with the NDE measurements (example shown in Figures 3.3 and 3.4), data from OLM sensors can further be used toward developing a DT framework. The OLM data can be from laboratory tests or from measurements made directly on-site. More data obtained from different sources can help improve the predictive capability of a DT system. Figure 3.5 shows representative data from laboratory test using active ultrasonic sensor. The signals are associated with the measurements made at a given instance in time. In this case, a U-bend SG tube was interrogated in real time (online) in order to monitor SCC initiation and growth [26]. Figure 3.6 shows the corresponding estimated flaw sizes (damage indices) as a function of time. Figure 3.7 shows an image of the U-bend tube after termination of the SCC manufacturing process. In addition to dealing with data from different sources (data heterogeneity), another major complexity that a DT database system is faced with is the variability in time scales. For example, the OLM measurements mentioned above can be of very small-time scale (e.g., in milliseconds) compared to the other types of sensor measurements. Figure 3.8 shows an example of laboratory test-based data that can be used for DT model development. This figure shows the fatigue cycle versus observed cyclic strain under thermal-mechanical fatigue loading [28]. Although the laboratory data presented here were obtained using a truncated cyclic period (of approximately 25 seconds), actual reactor loading cycles could vary from days to months or even more than a year (e.g. refer Figure 2.7 for design-basis fuel cycle). For such variable time scales of sensor measurements, the chosen database type must have the flexibility to accommodate different length scale data. The chosen database must have flexibility not only to add new data sources (those might also need to be associated with new data types, e.g. point sensor data versus 2-d image data, etc.), but also must have the flexibility for allowing horizontal scaling of the database by adding new storage hardware (for example to deal with saving large amount of long-term sensor data from nuclear power plants those can operate beyond 60-80 years or more [29]).

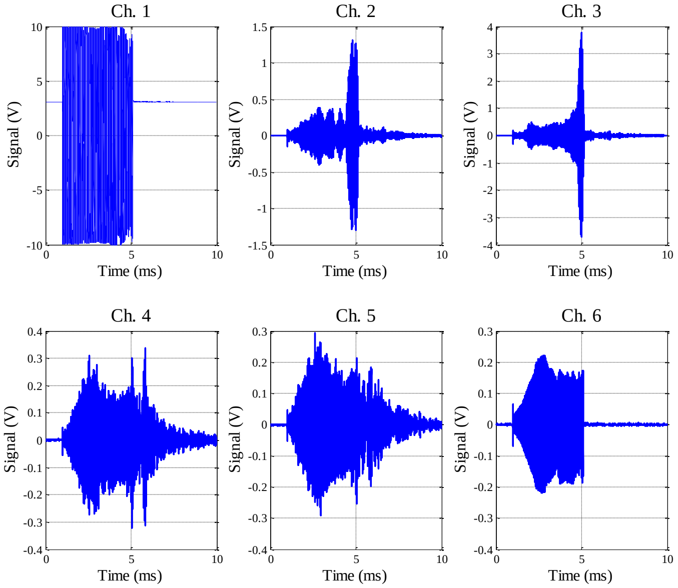


Figure 3. 5 Example of laboratory test based active ultrasound signals from a SG U-bend tube at a given instance while the tube being interrogated in real time (online) for a stress corrosion cracking testing. Figure shows sample signal from actuator (top left), sensor group 1 (top center and top right), sensor group 2 (bottom left and bottom center), and noise sensor (bottom right) [26].

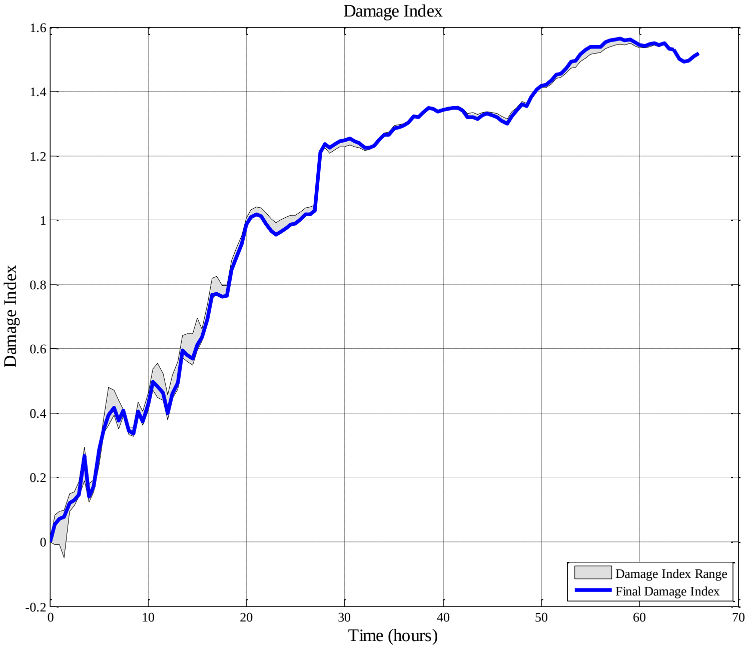


Figure 3. 6 Example of time versus estimated flaw size (damage index) of the SG U-bend tube interrogated with an online active ultrasound technique [26].

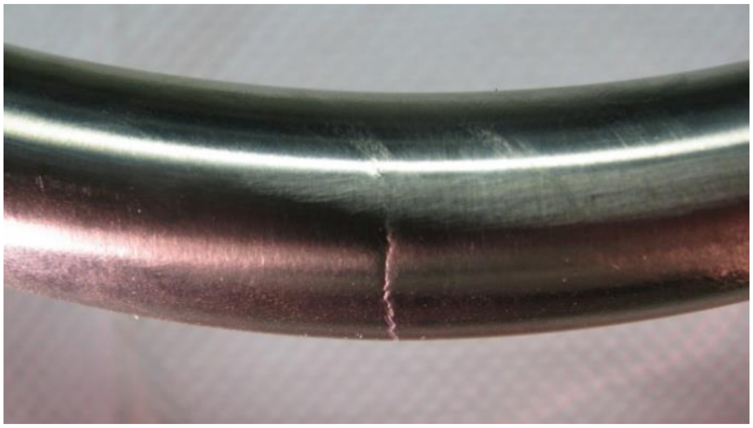


Figure 3. 7 A picture of the U-bend tube specimen after termination of the SCC test [26].

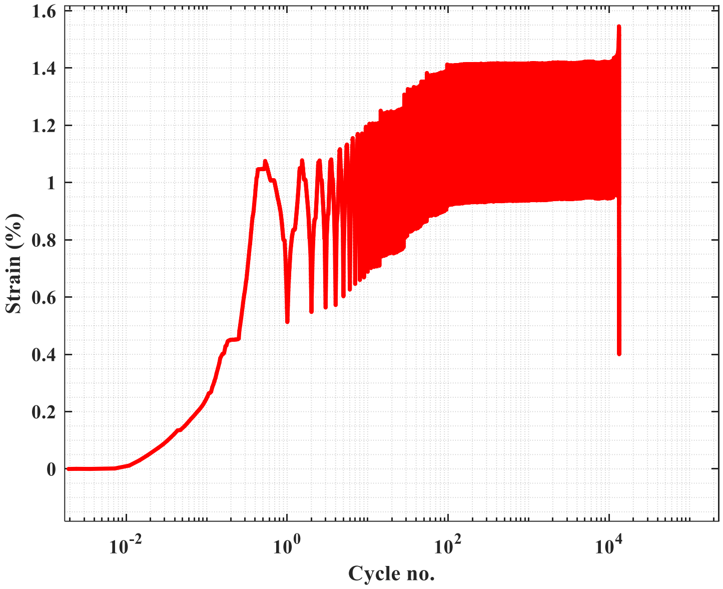


Figure 3. 8 Example of laboratory test-based measured cycle versus observed strain under fatigue loading, with cyclic period of 25 seconds [28]. The time scale of measurements can be significantly different from the time scale for ultrasound sensor measurements shown in Figure 3.5.

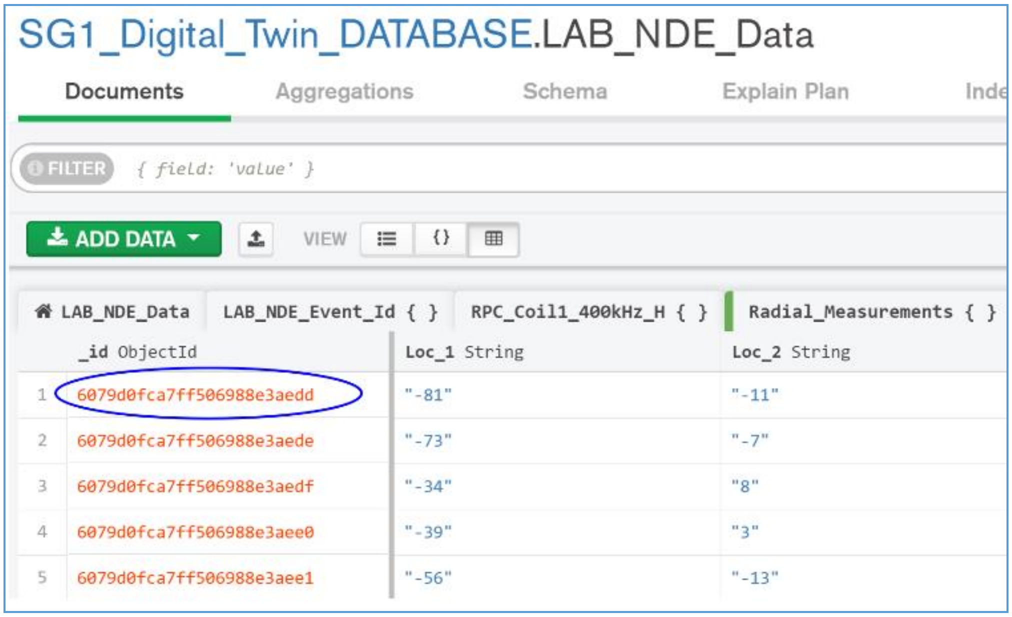
3.3 MongoDB Based Database and PyMongo Based Database Interface Application

In the previous section, we presented some sample data for DT applications. In this section we discuss storing of those data to a MongoDB-based NoSQL database for future use in a DT predictive analytics framework. The goal is to ultimately develop a DT framework for nuclear reactor heat-exchangers (e.g., steam generator used in a pressurized water reactor and/or heat-exchangers for boiling water reactor), with hundreds of subcomponents such as u-bend tubes (refer to Figure 2.1, which shows an example u-bend tube in a steam generator), that transfer the heat between primary loop (which directly produces the heat through nuclear chain reaction) and secondary loop (which transfers the primary-loop heat to steam that drives the turbine). An example of MongoDB based database implementation with storing of data from various sources: such as from virtual sensor measurements (from finite element-based heat transfer model) at thousands of 3D nodal locations (or point-clouds) and real sensor measurements (from continuous OLM sensors and from non-continuous NDE measurements) are presented here. Although at present we only show representative data from laboratory measurements (from both OLM and NDE sensors), the eventual goal is to include sensor measurements from actual onsite NPP components. The database also needs to be linked to a Python language-based DT predictive analytics framework. This is attained through a PyMongo based database interface application [30, 31]. Earlier shown Figure 1.3 shows the schematic diagram for the flow of data in the proposed DT framework, which is currently under development. Shown next are some test case results related only to MongoDB based database implementation and data querying and visualization using PyMongo interface application.

3.3.1 Example MongoDB Database

Sensor measurements from OLM and NDE sensors can be used for estimating the flaw size in a component at a given time. Figure 3.9 shows representative nested NDE data that are imported to MongoDB database. The laboratory test in this case was conducted on a steam generator tube subjected to SCC testing. These types of data can be used for developing the current-state-estimator leg of the DT framework. Figure 3.10 shows the nested data structure of the highlighted data object shown in Figure 3.9. The eddy current measurements were made using a rotating probe with different coils, generating 3D data. This means each sensor measurement points correspond to a circumferential position and axial position along the SG tube and the corresponding signal components. Also, each sensor measurement corresponds to a particular frequency which has both real and imaginary parts, or in other words vertical (V) and horizontal (H) components. The example data shown in Figures 3.9 and 3.10 correspond to a coil at a frequency of 400 kHz. The figures only show the vertical components of the measured signal. Nevertheless, this example shows the complexity of the data hierarchy that can easily overwhelm a conventional RDBMS type database with rigid schema and fixed data structure. However, by using a MongoDB type NoSQL database, these types of complex data can be stored as a document with nested key-value attributes and can be easily pulled out using appropriate query.

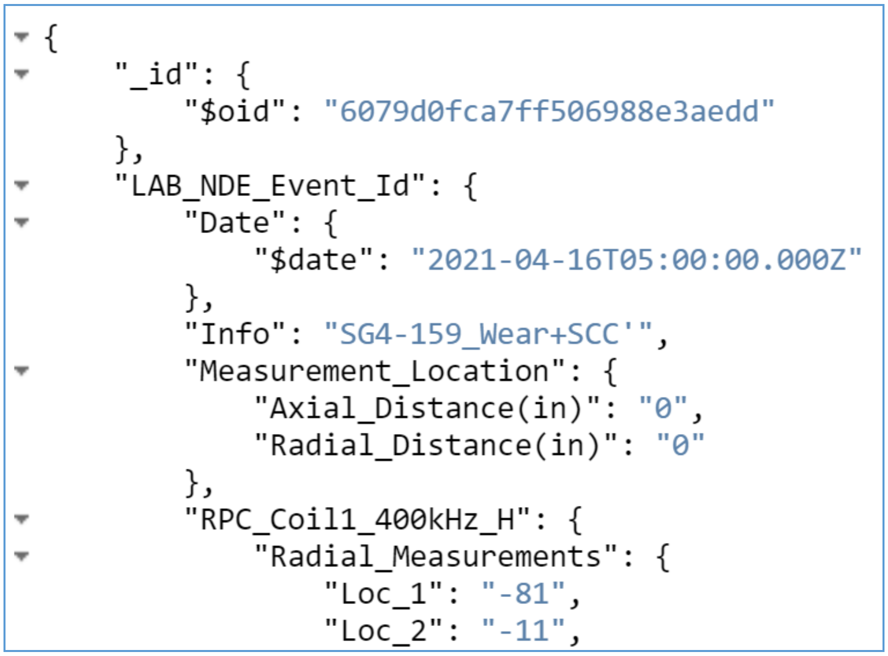
Figure 3.11 shows representative FE process input data those were imported to MongoDB database. These types of data can be used for developing the future-state-predictor leg of the DT framework. Figure 3.12 shows the data structure of the highlighted data object shown in Figure 3.11. These figures show seven different fields some of which were used as processes inputs for FE based virtual sensor data prediction. For example, these fields are time (within a given fuel cycle), the inlet, outlet, steam and feed water temperatures and the primary and secondary loop pressure inputs. These types of process input data can be used for developing a mapping function with respect to FE simulated nodal temperature (refer sections 4 to 7). In actual reactor implementation of the DT model, for a given set of above-mentioned process inputs (which can be measured in real time), the corresponding nodal temperature can be predicted. Note that information about nodal temperature spatial-temporal distribution is required to accurately estimate the corresponding thermal strain and at the same time the corresponding stress states that drive the actual damage growth. Figure 3.13 shows the example FE simulated virtual sensor data those were imported to the MongoDB database. As discussed above, these types of data can be used for developing the future-state-predictor leg of the DT framework. Figure 3.13 shows 3720 nodal temperature fields at 3720 spatial point-clouds or FE nodes (node ID 23382 to 27101) of a single SG tube (refer Figure 2.15), with each field representing a time-series data. Figure 3.14 shows the data structure of the highlighted data object shown in Figure 3.13. If the data from all the tubes within the SG are to be considered, there can be millions of nodal locations with each location having different time-series of temperatures. This example shows the importance of selecting a appropriate database that can not only allow adding new data types (e.g. for future application needs), but also allow future storage augmentation (e.g. of horizontal scaling of an existing database using cheaper commodity servers).



SG1_Digital_Twin_DATABASE.LAB_NDE_Data

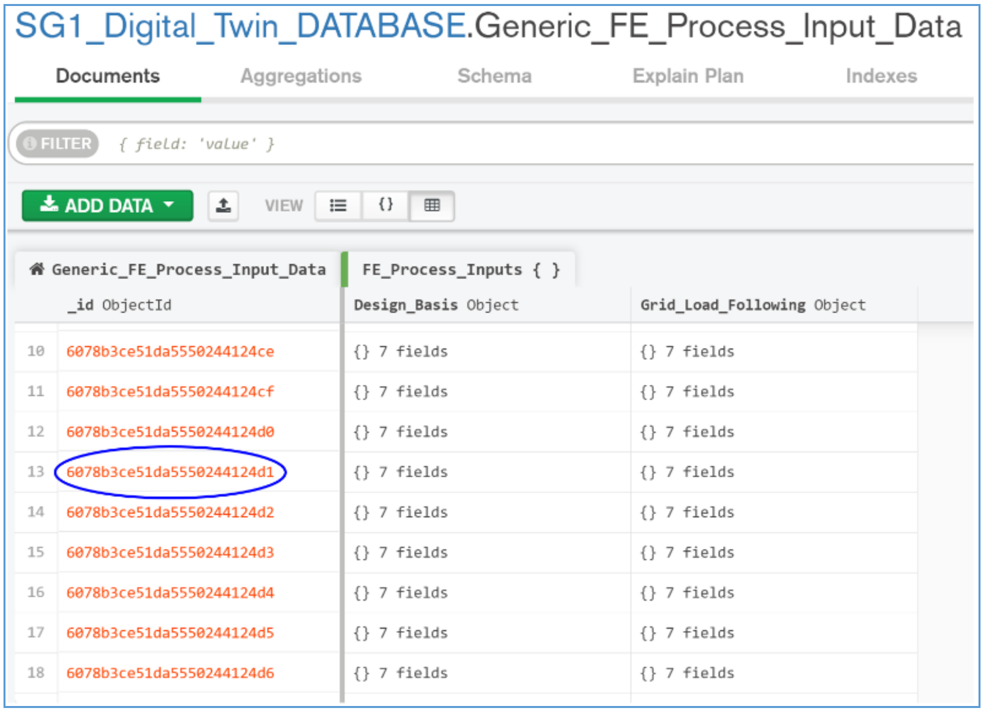
	_id ObjectId	Loc_1 String	Loc_2 String
1	6079d0fca7ff506988e3aedd	"-81"	"-11"
2	6079d0fca7ff506988e3aede	"-73"	"-7"
3	6079d0fca7ff506988e3aedf	"-34"	"8"
4	6079d0fca7ff506988e3aee0	"-39"	"3"
5	6079d0fca7ff506988e3aee1	"-56"	"-13"

Figure 3. 9 Example laboratory test based nested NDE data those were imported to MongoDB database.



```
{
    "_id": {
        "$oid": "6079d0fca7ff506988e3aedd"
    },
    "LAB_NDE_Event_Id": {
        "Date": {
            "$date": "2021-04-16T05:00:00.000Z"
        },
        "Info": "SG4-159_Wear+SCC'",
        "Measurement_Location": {
            "Axial_Distance(in)": "0",
            "Radial_Distance(in)": "0"
        },
        "RPC_Coil1_400kHz_H": {
            "Radial_Measurements": {
                "Loc_1": "-81",
                "Loc_2": "-11",
```

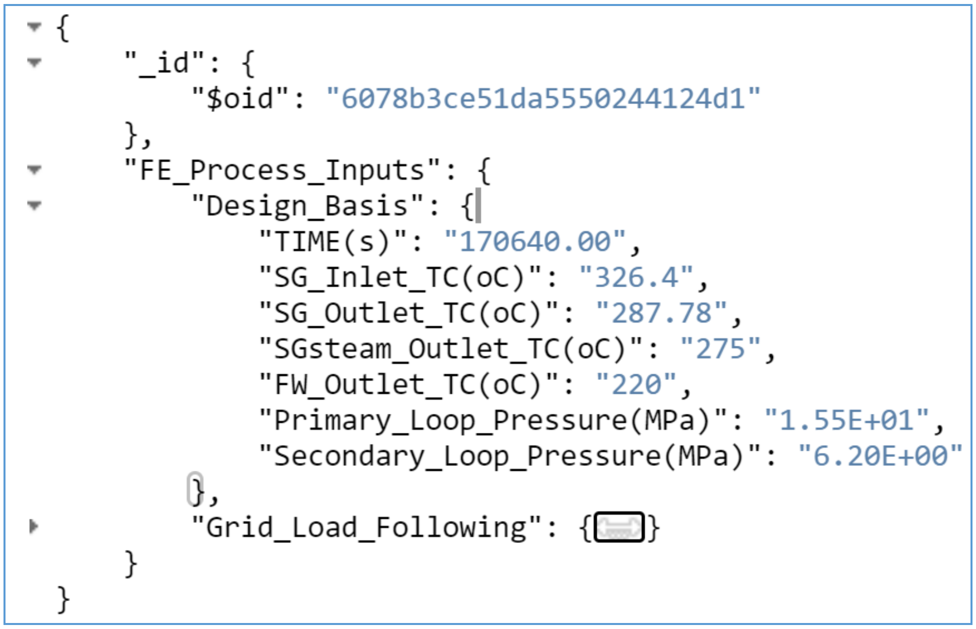
Figure 3. 10 Data structure of the highlighted data object in Figure 3.9.



SG1_Digital_Twin_DATABASE.Generic_FE_Process_Input_Data

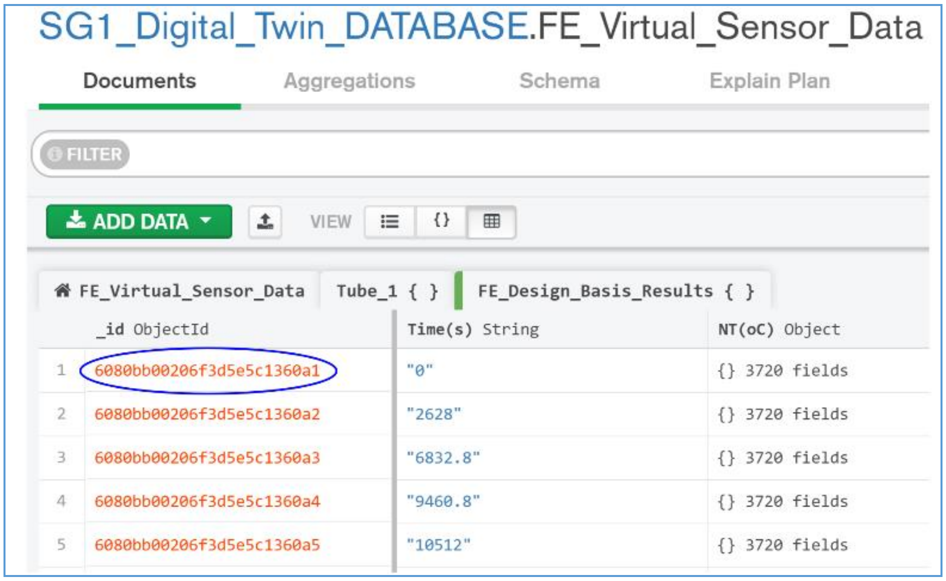
	_id ObjectId	Design_Basis Object	Grid_Load_Following Object
10	6078b3ce51da5550244124ce	{} 7 fields	{} 7 fields
11	6078b3ce51da5550244124cf	{} 7 fields	{} 7 fields
12	6078b3ce51da5550244124d0	{} 7 fields	{} 7 fields
13	6078b3ce51da5550244124d1	{} 7 fields	{} 7 fields
14	6078b3ce51da5550244124d2	{} 7 fields	{} 7 fields
15	6078b3ce51da5550244124d3	{} 7 fields	{} 7 fields
16	6078b3ce51da5550244124d4	{} 7 fields	{} 7 fields
17	6078b3ce51da5550244124d5	{} 7 fields	{} 7 fields
18	6078b3ce51da5550244124d6	{} 7 fields	{} 7 fields

Figure 3. 11 Example FE process input (which also can be measured in an actual NPP) data those were imported to MongoDB database.



```
{
    "_id": {
        "$oid": "6078b3ce51da5550244124d1"
    },
    "FE_Process_Inputs": {
        "Design_Basis": {
            "TIME(s)": "170640.00",
            "SG_Inlet_TC(oC)": "326.4",
            "SG_Outlet_TC(oC)": "287.78",
            "SGsteam_Outlet_TC(oC)": "275",
            "FW_Outlet_TC(oC)": "220",
            "Primary_Loop_Pressure(MPa)": "1.55E+01",
            "Secondary_Loop_Pressure(MPa)": "6.20E+00"
        },
        "Grid_Load_Following": {}
    }
}
```

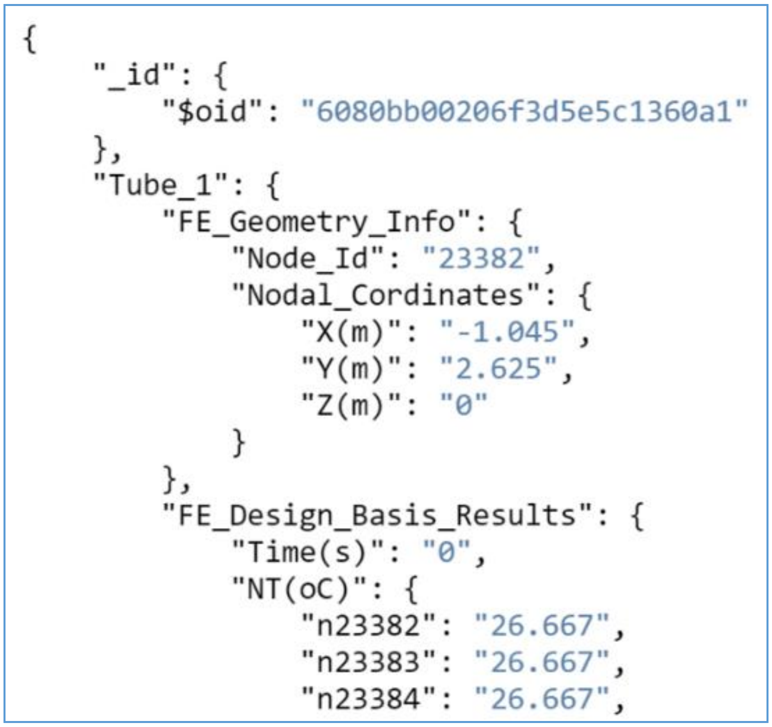
Figure 3. 12 Data structure of the highlighted data object in Figure 3.11.



SG1_Digital_Twin_DATABASE.FE_Virtual_Sensor_Data

	_id ObjectId	Time(s) String	NT(oC) Object
1	6080bb00206f3d5e5c1360a1	"0"	{} 3720 fields
2	6080bb00206f3d5e5c1360a2	"2628"	{} 3720 fields
3	6080bb00206f3d5e5c1360a3	"6832.8"	{} 3720 fields
4	6080bb00206f3d5e5c1360a4	"9460.8"	{} 3720 fields
5	6080bb00206f3d5e5c1360a5	"10512"	{} 3720 fields

Figure 3. 13 Example FE virtual sensor data (nodal temperature time-series at 3720 nodal locations or point-clouds), those were imported to MongoDB database.



```
{
    "_id": {
        "$oid": "6080bb00206f3d5e5c1360a1"
    },
    "Tube_1": {
        "FE_Geometry_Info": {
            "Node_Id": "23382",
            "Nodal_Cordinates": {
                "X(m)": "-1.045",
                "Y(m)": "2.625",
                "Z(m)": "0"
            }
        },
        "FE_Design_Basis_Results": {
            "Time(s)": "0",
            "NT(oC)": {
                "n23382": "26.667",
                "n23383": "26.667",
                "n23384": "26.667",
```

Figure 3. 14 Data structure of the highlighted data object in Figure 3.13.

3.3.2 Example Results: PyMongo Application Interface Based Data Query and Visualization

The data presented above have to eventually be used for predictive analytics applications (refer schematic shown in Figure 1.3). We are developing a Python language based predictive analytics framework to predict the future time-series states of the component based on which to the state of the component in the future can be predicted. Some of the theoretical background and related results follows (refer sections 4 to 7). Nevertheless, an appropriate database interface must be developed for the passing of the data from the database to a DT predictive analytics framework. We have developed a PyMongo [30, 31] application interface to query and visualize the required data before sending it into the DT predictive analytics framework. Figure 3.15 shows the PyMongo application interface queried virtual sensor measurements at all the 3D nodes or point-clouds shown in Figure 2.15. Figure 3.17 shows the example scatter plot of the nodal temperatures (shown in Figure 3.15) at a typical time.

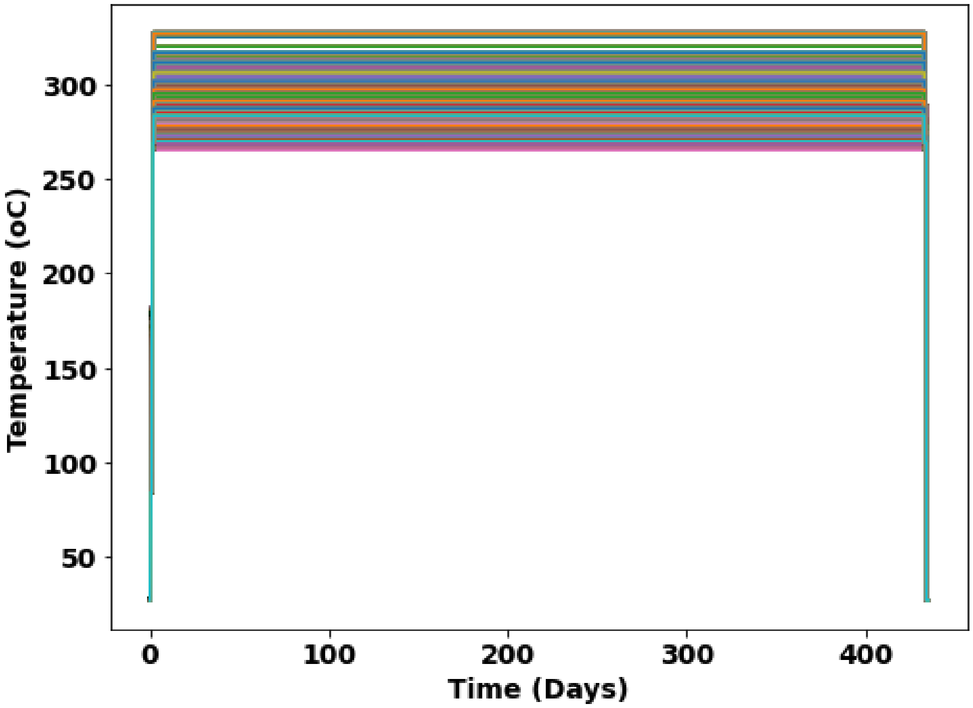


Figure 3. 15 PyMongo application interface queried virtual sensor measurements at all the 3D nodes (or point-clouds) shown in Figure 2.15.

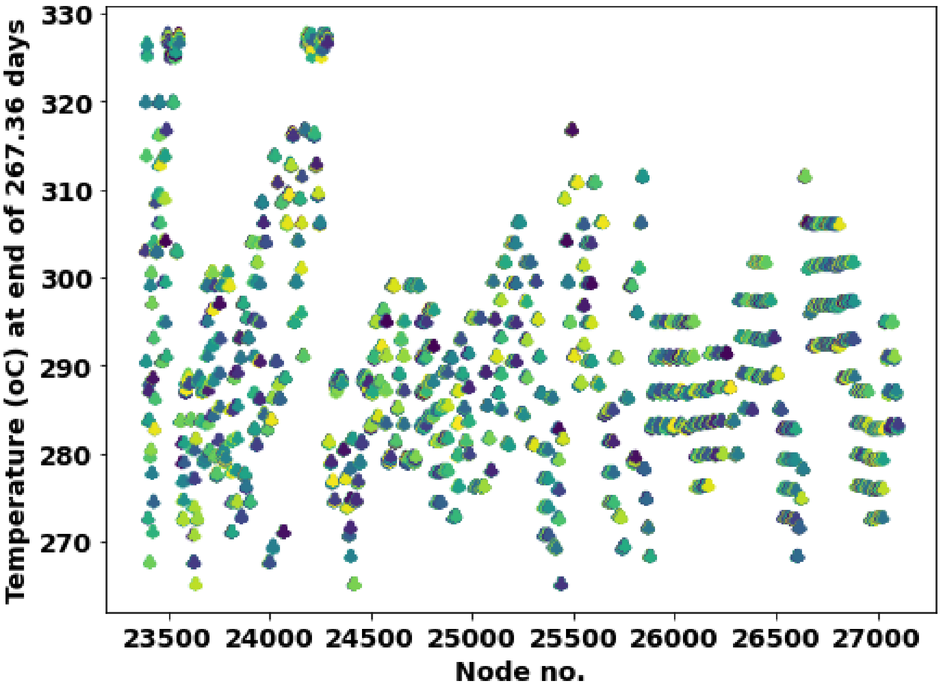


Figure 3. 16 Example scatter plot of the nodal or point cloud temperatures (shown in Figure 3.15) at a typical time.

4 Theoretical Background of Different AI/ML Predictive Models

The traditional way of predicting spatial-temporal variation of temperature is through FE model-based heat transfer analysis. Although, the FE based approach is a well-established approach for component temperature prediction, it has four major drawbacks: a) each change in input/process thermal boundary conditions requires to rerun the heat transfer analysis, b) simulating for longer duration (e.g. under multi-year-multi-cycle reactor loading) is not only computationally time consuming but also there will be issues of storing large amount of FE simulated result database, c) requires specific skill to perform heat transfer analysis, d) FE models are offline predictive tools, not suitable for online/real-time applications. These drawbacks can be a hinderance for implementing FE based digital-twin technology for real-time state prediction of reactor components. To tackle this issue, we are developing a digital-twin framework, which is based on AI/ML based multi-time-series predictive models. The data-driven AI/ML models need to be trained based on the process input versus geometry-dependent temperature data. These training data must be obtained from a one-time FE based heat-transfer analysis. For predicting temperature at a future time, the proposed AI/ML based expert elicitation system would only require the information of few input/process variables (e.g., time-dependent primary loop pressure, temperature, etc.) measured from a few existing sensors, without the need of installing any additional sensor systems. Note that installing new sensors is a major hindrance for rapid digitalization and implementation of AI/ML technologies in nuclear sector. This is not only due to highly regulated nuclear industry, but also due to inaccessible locations and high radiation environment for sensor placements. Towards this goal different AI/ML predictive models (for spatial-temporal temperature predictions) were developed, and their performance were evaluated. The resulting model should be robust for any given input/process conditions without being overfitted. Four type of AI/ML regression techniques such as linear regression, decision tree based random forest and gradient-boosting and kernel function-based support-vector-machine regression models were evaluated. These techniques were implemented through python language based scikit-learn [32] AI/ML libraries. The mathematical backgrounds of these AI/ML regression techniques are briefly discussed below.

4.1 Multivariate Linear (LR) Regressor

The time-dependent temperatures (or output or target variables) at any spatial-location (k), can be predicted as a linearly dependent function of few input/process variables those were only measured through a few existing sensors (without the need of installing any additional sensor systems). With multiple process measurements as input variables, and the requirement of predicting geometry-dependent temperature at multiple locations (along all the 3D directions of the component material) can lead to a Multi-Input-Multi-output (MIMO) predictive models. Figure 4.1 shows the schematic of a MIMO time-series and associated input-output matrices. The temperature time-series $\hat{y}_{i,k}$ at any given time ($\boldsymbol{t_{i=1,\dots m}}$) and spatial-location ($k = 1, \dots, o$) can be predicted using the following expression:

$$\hat{y}_{i,k} = w_{k,0} + w_{k,1}x_{i,1} + w_{k,2}x_{i,2} + \cdots + w_{k,j}x_{i,j} + \cdots + w_{k,n}x_{i,n} \qquad (4.1)$$

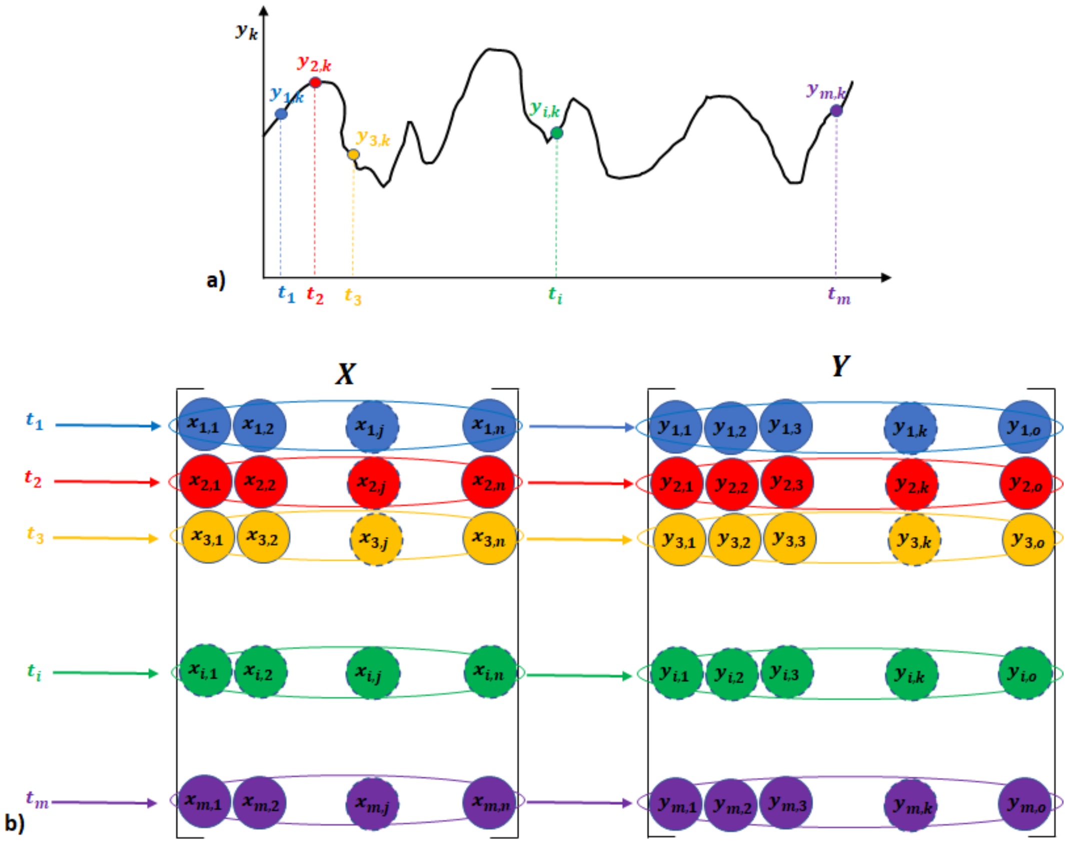
In Eq. (4.1), $w_{k,j}$ are the weights, $x_{i,j}$ are the different process/input variables with '$i = 1,2,\cdots,m$', '$j = 1,2,\cdots,n$', and '$k = 1,2,\cdots,o$' represent the indices for time, process/input variables, and output or spatial locations, respectively. The $o \times n$ wight matrix $\boldsymbol{W} = f(w_{k,j})$ must be estimated given the training or historical input-output $(\boldsymbol{X} \rightarrow \boldsymbol{Y})$ data set. The model parameters or the weights must be estimated only once and based on the FE based heat transfer analysis results. Then the resulting trained model can be used in real-time (without the need of conducting any offline FE simulation) to predict the geometry and time dependent temperature for a given time-dependent process measurements. The accuracy of each prediction is evaluated using two criterion such as mean square error (MSE_k) and coefficient of determination (R^2), which respective expressions are as given below:

$$MSE_k = \frac{1}{m}\sum_{i=1}^{i=m}(y_{i,k} - \hat{y}_{i,k})^2 \tag{4.2}$$

$$R_k^2 = 1 - \frac{u}{v} = \quad 1 - \frac{\sum_{i=1}^{i=m}(y_{i,k}-\hat{y}_{i,k})^2}{\sum_{i=1}^{i=m}(y_{i,k}-\bar{y}_k)^2} \tag{4.3}$$

In Eq. (4.2), $y_{i,k}$ and $\hat{y}_{i,k}$ are respectively the true and predicted temperature at time t_i and location k, whereas in Eq. (4.3) $\bar{y}_k$ represents the mean of all true observations at location k and is expressed as below:

$$\bar{y}_k = \frac{1}{m}\sum_{i=1}^{i=m} y_{i,k} \tag{4.4}$$



a)
y_k
$y_{1,k}$
$y_{2,k}$
$y_{3,k}$
$y_{i,k}$
$y_{m,k}$
t_1 t_2 t_3 t_i t_m
b)
X
Y
t_1 $x_{1,1}$ $x_{1,2}$ $x_{1,j}$ $x_{1,n}$ $y_{1,1}$ $y_{1,2}$ $y_{1,3}$ $y_{1,k}$ $y_{1,o}$
t_2 $x_{2,1}$ $x_{2,2}$ $x_{2,j}$ $x_{2,n}$ $y_{2,1}$ $y_{2,2}$ $y_{2,3}$ $y_{2,k}$ $y_{2,o}$
t_3 $x_{3,1}$ $x_{3,2}$ $x_{3,j}$ $x_{3,n}$ $y_{3,1}$ $y_{3,2}$ $y_{3,3}$ $y_{3,k}$ $y_{3,o}$
t_i $x_{i,1}$ $x_{i,2}$ $x_{i,j}$ $x_{i,n}$ $y_{i,1}$ $y_{i,2}$ $y_{i,3}$ $y_{i,k}$ $y_{i,o}$
t_m $x_{m,1}$ $x_{m,2}$ $x_{m,j}$ $x_{m,n}$ $y_{m,1}$ $y_{m,2}$ $y_{m,3}$ $y_{m,k}$ $y_{m,o}$

Figure 4. 1 Schematic of a) Temperature time-series ($\boldsymbol{y}_k$) at any spatial-location (k), b) MIMO representation of the input-output matrix with respect to different time ($\boldsymbol{t}_{i=1,\ldots m}$).

4.2 Decision-Tree Model

The traditional linear regression model might produce wrong results possibly due to noisy and overfitting of training data. Hence, it is essential to check other type of predictive model and compare the performance against the linear model discussed above. For the purpose, different decision-tree based AI/ML models were evaluated. In general decision-tree methods are non-parametric supervised learning methods, which depends on flowchart like tree (decision-tree) structure with different nodes such as root, internal and leaf nodes. Note that, nonparametric models are not based solely on parametrized families of probability distributions (e.g., the mean and variance), restricting the use of the predictive model for any future situations with data following different probability distributions. Figure 4.2 shows the schematic of a single-variate (e.g. dependent only on input feature $\boldsymbol{x}_{\boldsymbol{i=1,\ldots m,\,j}}$) decision tree to predict the temperature time-series ($\boldsymbol{y}_{i=1,\ldots m,\,k}$) at a spatial-location of k. There are different types of decision tree algorithms. Out of those, ID3 (Iterative Dichotomiser 3) is one of the earliest developed decision tree algorithms, which was developed during late 1970s and early 1980 by machine learning researcher Ross Quinlan [33]. An advanced version of the ID3 popularly known as C4.5 was later presented by Quinlan. Around the same time the CART (Classification and Regression Trees) decision tree algorithm which is very similar to C4.5 was independently presented by group of statisticians L. Breiman, J. Friedman, and R. Olshen [34]). In all the above-mentioned algorithms, decision trees are constructed in a top-down recursive divide and conquer manner. Nevertheless, first the tree splitting procedure has to be trained based on a set of historical or training data (which include training features and the associated class/label or target values) and attribute selection methods. Different machine learning libraries (those are currently being popular and widely being used in various industrial sectors such as banking, social media, driverless car, etc.) are broadly based on original ID3 or C4.5 or CART type algorithms but with some improved algorithms. For example, the scikit-learn [32] uses an optimized version of the CART algorithm which constructs binary trees using the feature and threshold that yield the largest information gain at each node. We have used the scikit-learn based libraries in the discussed work. The mathematical background behind the scikit-learn [32] based decision tree library is briefly presented below.

Given the set of training data $D = \left\{\boldsymbol{x}_{\boldsymbol{i=1,\ldots m,j}}, \boldsymbol{y}_{\boldsymbol{i=1,\ldots m,k}}\right\}$, the feature or input space can be partitioned such as way the training samples with same labels (for classification) or targets (for regression) can be grouped together. At each node (say at node p) the presplit data set (say D_p) must be binary separated either to the left or right side of that node. The split is based on split parameter, $\theta = (x_j, x_j^p)$ which is pertinent to a particular feature x_j and the corresponding threshold x_j^p at that node (p). The optimized split parameter θ^* can be estimated by minimizing the following impurity function:

$$\theta^* = argmin_\theta I(D_p, \theta) \tag{4.5}$$

In Eq. 4.5, the impurity function can be defined as:

$$I\left(D_p, \theta\right) = \frac{N_p^{left}}{N_p} L\left(D_p^{left}(\theta)\right) + \frac{N_p^{right}}{N_p} L(D_p^{right}(\theta)) \tag{4.6}$$

In Eq. 4.6, $L()$ represents the loss function, whereas N_p, N_p^{left}, and N_p^{right} are respectively the number of presplit, left-post-split and right-post-split samples at node p. The $D_p^{left}(\theta)$ and $D_p^{right}(\theta)$ are the split parameter (θ) dependent left and right sample sets and are as given as below:

$$D_p^{left}(\theta) = \left\{(x, y) | x_j \leq x_j^p\right\} \tag{4.7}$$

$$D_p^{right}(\theta) = D_p \backslash D_p^{left}(\theta) \tag{4.8}$$

The scikit-learn [32] based decision tree algorithm split recursively the subsets $D_p^{left}(\theta^*)$ and $D_p^{right}(\theta^*)$ until the maximum allowable depth is reached, $N_p < min_{samples}$ or N_p =1. In Eq. 4.6, the loss function $L()$ can be of different type. For the discussed work a MSE based loss function was used, which can be expressed as below:

$$L(D_p) = \frac{1}{N_p}\sum_{y\in D_p}(y-\bar{y}_p)^2$$

with $\bar{y}_p$ as:

$$\bar{y}_p = \frac{1}{N_p}\sum\nolimits_{y\in D_p} y \tag{4.9}$$

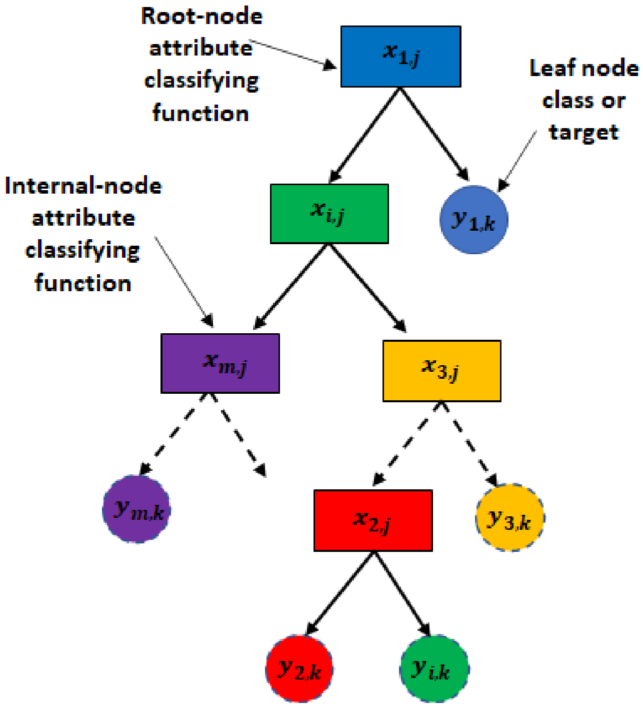

Root-node attribute classifying function
$x_{1,j}$
Leaf node class or target
Internal-node attribute classifying function
$x_{i,j}$
$y_{1,k}$
$x_{m,j}$
$x_{3,j}$
$y_{m,k}$
$x_{2,j}$
$y_{3,k}$
$y_{2,k}$
$y_{i,k}$

Figure 4. 2 Schematic of a single-variate (e.g., time-dependent input feature $\boldsymbol{x}_{i=1,\ldots m,\,j}$) decision tree to predict the temperature time-series ($\boldsymbol{y}_{i=1,\ldots m,\,k}$) at any spatial-location k.

4.3 Ensembled Decision-Tree Based Random-Forest (RF) Regressor

The base or single decision-tree based regression model might produce poor results associated with noisy data and overfitting. To improve accuracy in the single-tree based decision-tree based models various ensembled decision-tree methods have been invented [38-40]. This is through combining the effect of multiple decision-trees either horizontally or vertically. One of the popular examples of horizontally ensembled decision-tree based model is Random-Forest (RF) model [35] and vertically ensembled decision-tree based model is Gradient Boost (GB) model [36]. In the discussed work, we have used the RF and GB based ensembled decision-tree models. The brief mathematical background of the RF algorithm is discussed in this section, whereas the GB algorithm is discussed in the following section. The RF model is ensembled model with multiple decision-trees (refer Figure 4.3) which are horizontally assembled. A user must decide the number of trees to be modeled (subjected to the maximum possible combinations of underlying data sets).

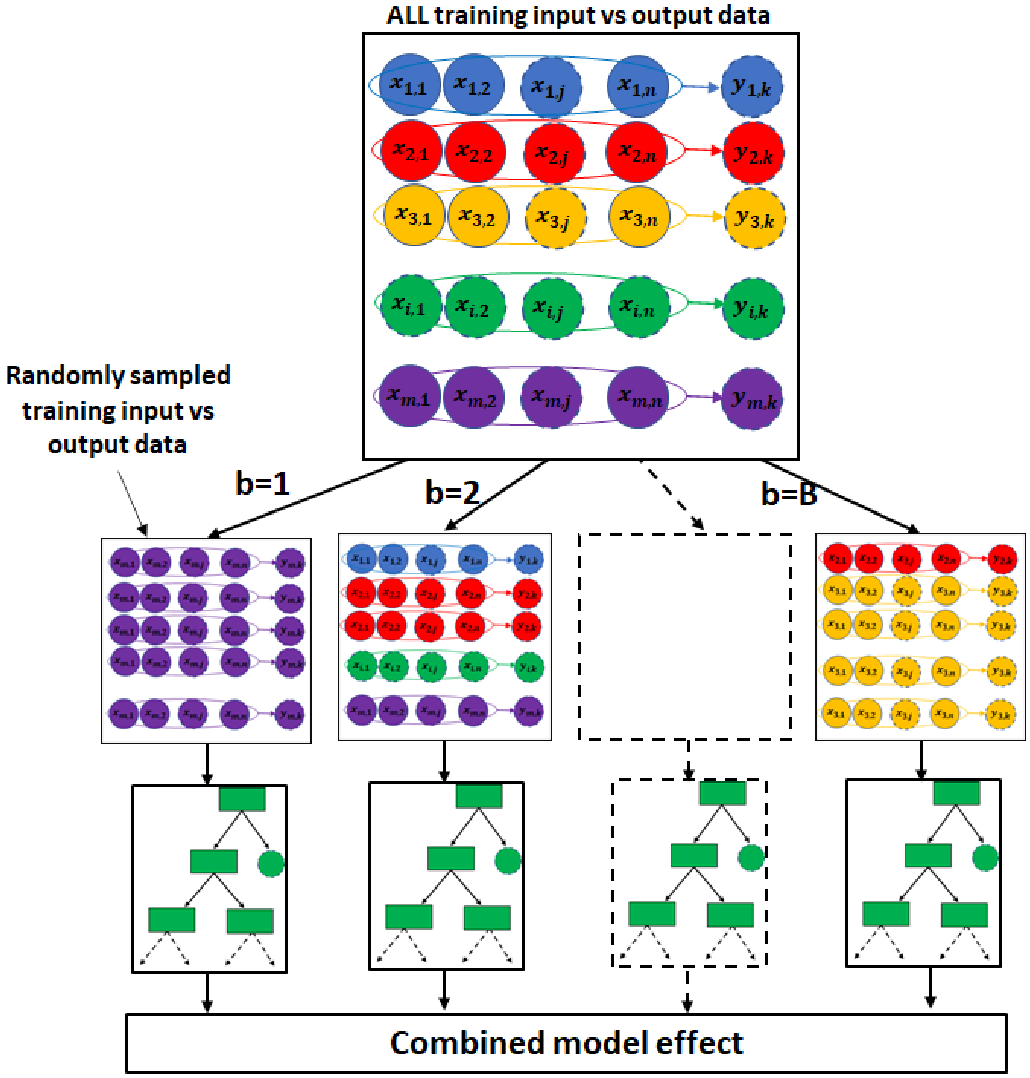

ALL training input vs output data
$x_{1,1}$ $x_{1,2}$ $x_{1,j}$ $x_{1,n}$ $y_{1,k}$
$x_{2,1}$ $x_{2,2}$ $x_{2,j}$ $x_{2,n}$ $y_{2,k}$
$x_{3,1}$ $x_{3,2}$ $x_{3,j}$ $x_{3,n}$ $y_{3,k}$
$x_{i,1}$ $x_{i,2}$ $x_{i,j}$ $x_{i,n}$ $y_{i,k}$
$x_{m,1}$ $x_{m,2}$ $x_{m,j}$ $x_{m,n}$ $y_{m,k}$
Randomly sampled training input vs output data
b=1
b=2
b=B
Combined model effect

Figure 4. 3 Schematic of a horizontally ensembled decision tree such as used in Random-Forest algorithm.

Each tree e.g., $b = 1, \ldots B$ has to be separately trained as discussed in the previous section. The RF model is based on the basic concept of bagging (bootstrap aggregating). Given 'm' tuples of original input-output data set $D = \{\boldsymbol{x}_{i=1,\ldots m,\, j=1,\ldots n}, \boldsymbol{y}_{i=1,\ldots m,\, k}\}$, the individual tree model $b = 1, \ldots B$, must be trained based on random selection of input-output data tuples. The randomly selected data set for the tree model 'b' can be referred as D_b. The data set D_b is randomly selected by randomly sampling the original input-output data set $D = \{\boldsymbol{x}_{i=1,\ldots m,\, j=1,\ldots n}, \boldsymbol{y}_{i=1,\ldots m,\, k}\}$ with replacements. Since sampling with replacement used, some of the original input-output tuples in D may not be included in D_b. In addition, some of the input-output tuples in D_b may be repeated (refer Figure 4.3 for illustration). Individual tree models to be trained separately using the randomly sampled data set $D_{b=1,\dots B}$ and using the procedure to train the single-tree model discussed in section 4.2. Once the individual trees are trained, for a given process/input tuple, the corresponding output to be predicted separately and the final results is the average of all the prediction. For example, the temperature prediction at any given time (t_i) and at the spatial location k can be predicted using the following expression:

$$\hat{y}_{i,k} = \frac{1}{B} \sum_{b=1}^{B} \hat{y}_{i,k}(D_b) \tag{4.10}$$

In Eq. (4.10) $\hat{y}_{i,k}(D_b)$ represents the prediction from tree model 'b' which was trained with randomly sampled data set D_b.

4.4 Ensembled Decision-Tree Based Gradient-Boosting (GB) Regressor

In addition to Random-Forest algorithm, the ensembled decision-tree based Gradient-Boosting (GB) algorithm was also evaluated for spatial- temporal temperature prediction. However, unlike the RF model, the GB model sequentially (or vertically) stacks multiple decision trees. The GB model pass the model prediction sequentially to the end node rather than parallel prediction as in case of RF model. Figure 4.4 shows the schematic of a vertically ensembled decision tree such as used in a typical GB algorithm. In a GB algorithm a single set of input-output data set $D = \left\{\boldsymbol{x}_{\boldsymbol{i}=\mathbf{1},\dots\boldsymbol{m},\, \boldsymbol{j}=\mathbf{1},\dots\boldsymbol{n}}, \boldsymbol{y}_{\boldsymbol{i}=\mathbf{1},\dots\boldsymbol{m},\, \boldsymbol{k}}\right\}$ is used without any random sampling as in case of RF algorithm. However, the use of a single set of input-output data set can lead to overfitting and erroneous prediction as in case of a typical single-tree based decision-tree model. However, the GB model uses stacks of individual decision-trees which sequentially reduces the prediction error. We have used the scikit-learn [32] based GB library, which mathematical backgrounds are briefly presented below.

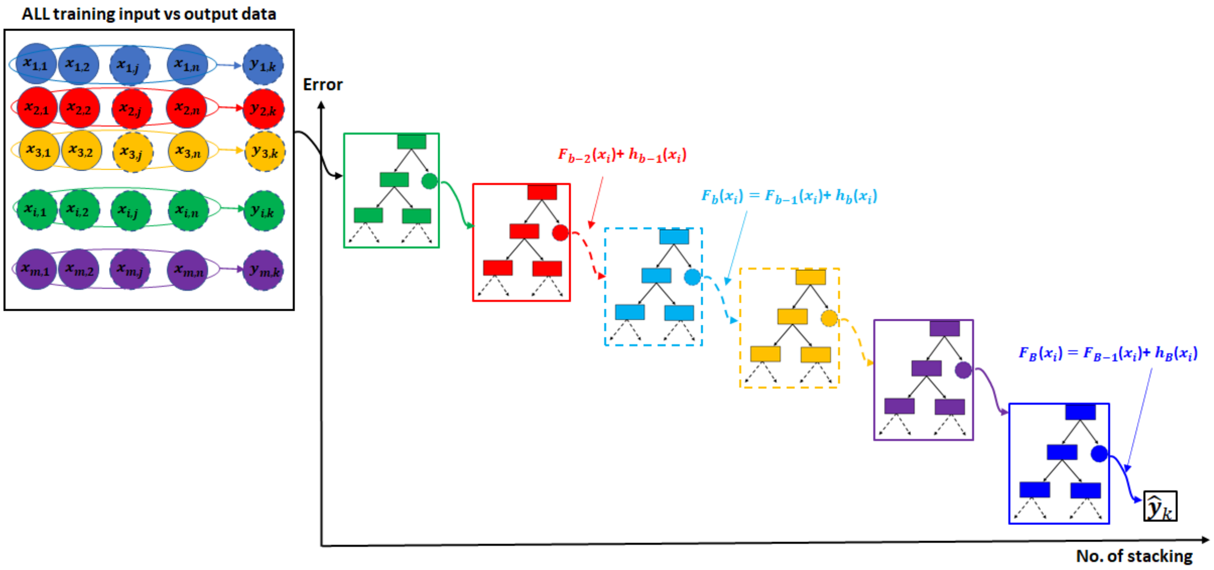

ALL training input vs output data
$x_{1,1}$ $x_{1,2}$ $x_{1,j}$ $x_{1,n}$ $y_{1,k}$
$x_{2,1}$ $x_{2,2}$ $x_{2,j}$ $x_{2,n}$ $y_{2,k}$
$x_{3,1}$ $x_{3,2}$ $x_{3,j}$ $x_{3,n}$ $y_{3,k}$
$x_{i,1}$ $x_{i,2}$ $x_{i,j}$ $x_{i,n}$ $y_{i,k}$
$x_{m,1}$ $x_{m,2}$ $x_{m,j}$ $x_{m,n}$ $y_{m,k}$
Error
$F_{b-2}(x_i)+ h_{b-1}(x_i)$
$F_b(x_i) = F_{b-1}(x_i)+ h_b(x_i)$
$F_B(x_i) = F_{B-1}(x_i)+ h_B(x_i)$
$\hat{y}_k$
No. of stacking

Figure 4. 4 Schematic of a vertically ensembled decision tree such as used in Gradient-Boosting algorithm.

Assuming there are $b = 1, \dots B$ stacked decision trees, the temperature at any given time (t_i) and spatial location k can be predicted using the following expression:

$$\hat{y}_{i,k} = F_B(x_i) = F_{B-1}(x_i) + h_B(x_i) \tag{4.11}$$

In Eq. (4.11), $F_B(x_i)$ represents the last decision-tree model in the sequence, whereas $F_{B-1}(x_i)$ and $h_B(x_i)$ represent the corresponding previous decision-tree model and the associated error function, respectively. Similarly, the model output at the end of tree model 'b' can be expressed as:

$$\hat{y}_{i,k,b} = F_b(x_i) = F_{b-1}(x_i) + h_b(x_i) \tag{4.12}$$

In Eq. (4.12), $F_b(x_i)$ represents the decision-tree model 'b' in the overall sequence, whereas $F_{b-1}(x_i)$ and $h_b(x_i)$ represent the corresponding previous decision-tree model and associated error function, respectively. At end of each decision-tree model 'b', the error function $h_b(x_i)$ can be evaluated using the following expression:

$$h_b(x_i) \approx \arg min_h \textstyle\sum_{i=1}^{m} h(x_i) g_i \tag{4.13}$$

In Eq. (4.13), $h_b(x_i)$ is minimized if $h(x_i)$ is fitted to predict a value that is proportional to the negative gradient of the samples (-g_i). The gradients are updated at each iteration by fitting the error function h_b for predicting the negative gradient of the samples.

4.5 High-Dimensional Kernel Function Based Support-Vector-Machine (SVM) Regressor

In addition to the above discussed linear model and ensembled decision-tree based RF and GB models, we have also evaluated different support vector regression (SVR) models [32, and 38] for predicting the spatial-temporal temperature. A SVR model, projects the nonlinear input or feature space to a high-dimensional space, where the transformed features can be linearly separable. This is through a chosen kernel function. Figure 4.5 shows the schematic showing the high-dimensional transformation of the input or feature space using a chosen kernel function ∅. In the high-dimensional feature space the transformed features can be linearly correlated with the outputs or the target values.

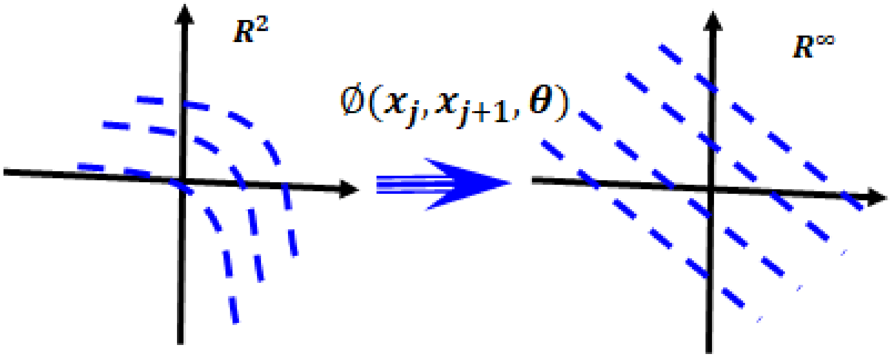

R^2
$\emptyset(\boldsymbol{x}_j, \boldsymbol{x}_{j+1}, \boldsymbol{\theta})$
R^∞

Figure 4. 5 Schematic showing the high-dimensional transformation of the input or feature space using kernel function ∅.

Using the SVR, the temperature at any given time t_i and spatial location k can be predicted using the following expression:

$$\hat{y}_{i,k} = (\boldsymbol{w}_j.\emptyset(\boldsymbol{x}_j, \boldsymbol{x}_{j+1})) + b_k \tag{4.14}$$

In Eq. (4.14) $\boldsymbol{w}_{j=1,\ldots n}$ and b_k are respectively the weight vector and a constant which has to be estimated by fitting the input-output data set $D = \{\boldsymbol{x}_{i=1,\ldots m,\, j=1,\ldots n}, \boldsymbol{y}_{i=1,\ldots m,\, k}\}$. Whereas $\emptyset(\boldsymbol{x}_j)$ are some chosen kernel function that transfer the input space to a high-dimensional space. In the discussed work three types of kernel functions such as constant, RBF (radial basis function) and sigmoid functions were evaluated. The expressions for the respective kernel functions are as below:

$$\emptyset_{linear} = < \boldsymbol{x}_j, \boldsymbol{x}_{j+1} > = \boldsymbol{x}_j.\boldsymbol{x}_{j+1} \tag{4.15}$$

$$\emptyset_{RBF} = \exp\left(-\gamma||\boldsymbol{x}_j - \boldsymbol{x}_{j+1}||^2\right) \tag{4.16}$$

$$\emptyset_{sigmoid} = \tanh\left(\gamma < \boldsymbol{x_j}, \boldsymbol{x_{j+1}} > +c\right) = \tanh\left(\gamma \boldsymbol{x_j}.\boldsymbol{x_{j+1}} + c\right) \tag{4.17}$$

In Eq.(4.16) $\gamma = 1/m\sigma^2$ with 'm' as the number of samples and 'σ^2' is the variance of samples. Whereas in Eq. (4.17) c is a specified constant such as equal to 0.

5 Multi-Time-Series Temperature Prediction in a SG-Tube: Evaluation of Different AI/ML Models with Few Outputs Under a Design-Basis Loading Cycle

Different AI/ML models described in section 4 were evaluated for their accuracy. The model evaluations were performed by predicting the temperature at only two random locations of the SG tube (refer Figure 2.1) and compared against the true/actual value simulated through the FE model discussed in section 2. Our assumption is, FE based heat transfer modeling procedures are well matured and produce accurate results. The main aim of developing the AI/ML based DT model is to predict the component states (e.g., temperature, stress, strain, etc.) in real time under any random process inputs which a FE model cannot. To note FE models are traditionally used offline for fixed set of underlying process boundary conditions. All the four AI/ML models (such as multivariate linear, Random-Forest, Gradient-Boost and Support-Vector-Machine) described in section 4 were evaluated using same set of input-output training data and validated against same set of input-output test data. Below are the related results.

5.1 Training & Test Data Used for the Evaluation Model

We considered the coolant-water temperature at SG inlet (which is the outlet of hot-leg) and SG outlet (which is the inlet of cold-leg) and at SG steam outlet and feedwater (FW) inlet are the four process or input variables. These input variables can affect the spatial-temporal temperature distribution in SG components (e.g. tube sheet, tubes, etc.) and in turn the stress-strain state (and associated damage progression) of a SG tube. We assume these input variables are measurable in an actual plant. We used FE data (simulated under a design-basis loading cycle) for training and testing the models discussed in this section. Portion of the related input-output data, mostly during the heat-up operation of the reactor were considered for training the model. Whereas, input-output data, during portion of the full-power operation and cool-down operation were considered for validating or testing the AI/ML models. Figure 5.1 shows the assumed process inputs covering an entire design-basis type loading cycle. Figure 5.2 shows the training portion of the input-temperature data (covering only during the heat-up operation of a design-basis type loading cycle). Whereas Figure 5.3 shows the corresponding test portion of the input-temperature data (covering portion of the steady-state power operation regime and the entire cool-down regime of a design-basis type loading cycle). Figure 5.4 shows the FE model simulated training-output-temperature data at two spatial locations. Overall, for all the discussed results in section 5, the AI/ML models had 4 inputs (or process variables) and 2 outputs (temperature at two spatial locations of SG tube) leading to a multi-input-multi-output (MIMO) prediction problem.

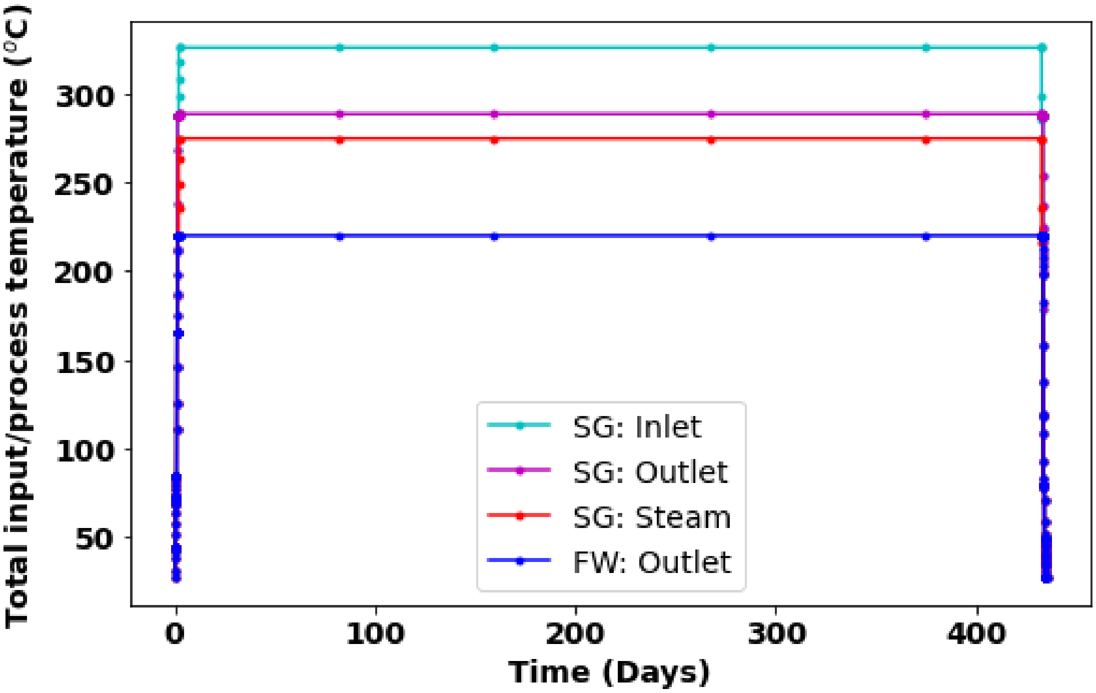


Figure 5. 1 Total-input-temperature data (covering an entire design-basis type loading cycle), which were used for training and validating all the discussed AI/ML models.

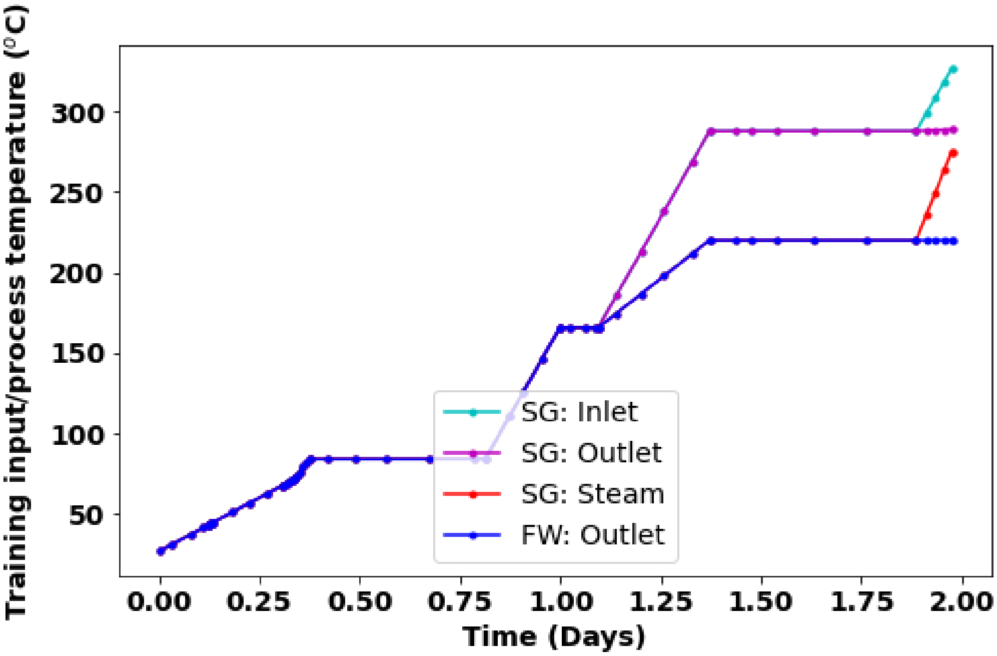


Figure 5. 2 Training-input-temperature data (covering only during the heat-up operation of a design-basis type loading cycle), which were used for training all the discussed AI/ML models.

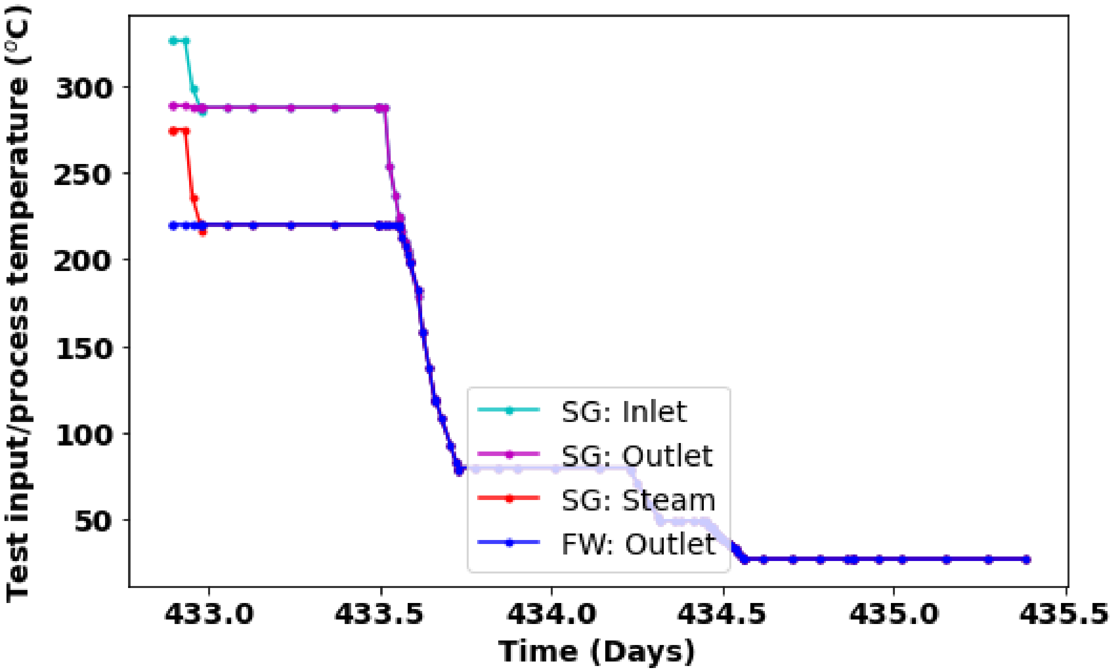


Figure 5. 3 Test-input-temperature data (covering portion of the steady-state power operation regime and the entire cool-down regime of a design-basis type loading cycle), which were used for validating all the discussed AI/ML models.

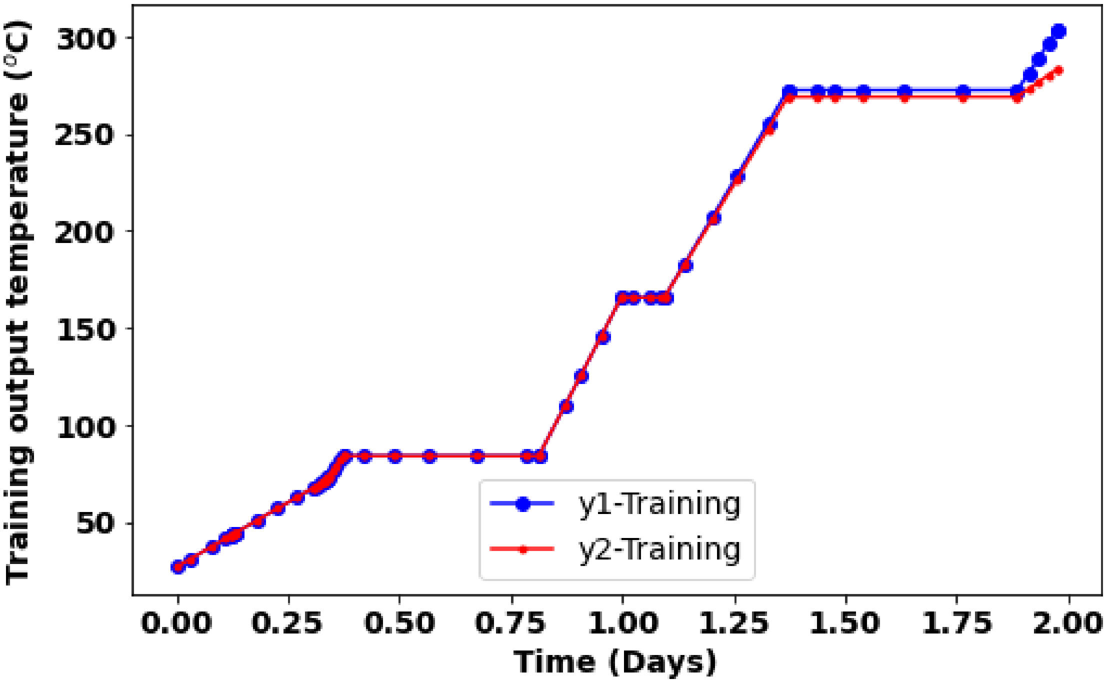


Figure 5. 4 FE model-based training-output-temperature data at two spatial locations (covering only during the heat-up operation of a design-basis type loading cycle), which were used for training all the discussed AI/ML models which results are presented in section 5.

5.2 Quantitative Evaluation of Multivariate Linear (LR) Regression Model

Using the training input-output data given in section 5.1, and AI/ML based multivariate regression technique discussed in section 4.1, a MIMO model was inferred. The related model evaluation results are presented here. For example, Figures 5.5 and 5.6 show different iterations versus training scores R2 (i.e., the unform average of coefficients of determination for training outputs y1 and y2), and example mean square error (for test output y1). Whereas Figures 5.7 and 5.8 show the actual values (simulated through FE model) versus AI/ML model predicted temperatures at nodal location 1 (i.e., output y1) and 2 (i.e., output y2), respectively. In firsthand look from Figures 5.7 and 5.8 there is good matching between AI/ML predicted temperature and actual (or FE predicted) temperature. Same is the case if we look at the training scores R2 results shown in Figure 5.5, that the coefficient of determination is very close to 1. However, these type of results (shown in Figure 5.5, 5.7 and 5.8) are not enough to judge the accuracy of the model. By checking the mean square error plot (shown in Figure 5.6) at certain randomized iteration the prediction error can be very high. These types of results made us to explore other type of AI/ML model such that the DT model can be robust for any randomized process inputs. The related model evaluation results follow.

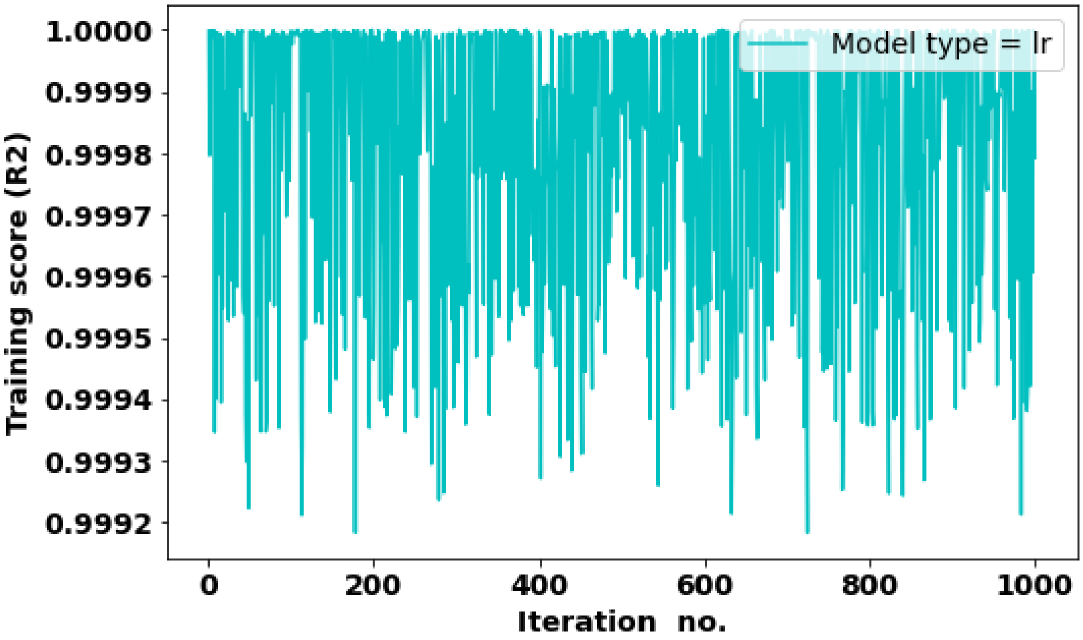


Figure 5. 5 Different iterations versus training scores R2 (i.e., the unform average of coefficients of determination for training outputs y1 and y2), estimated through linear regression model.

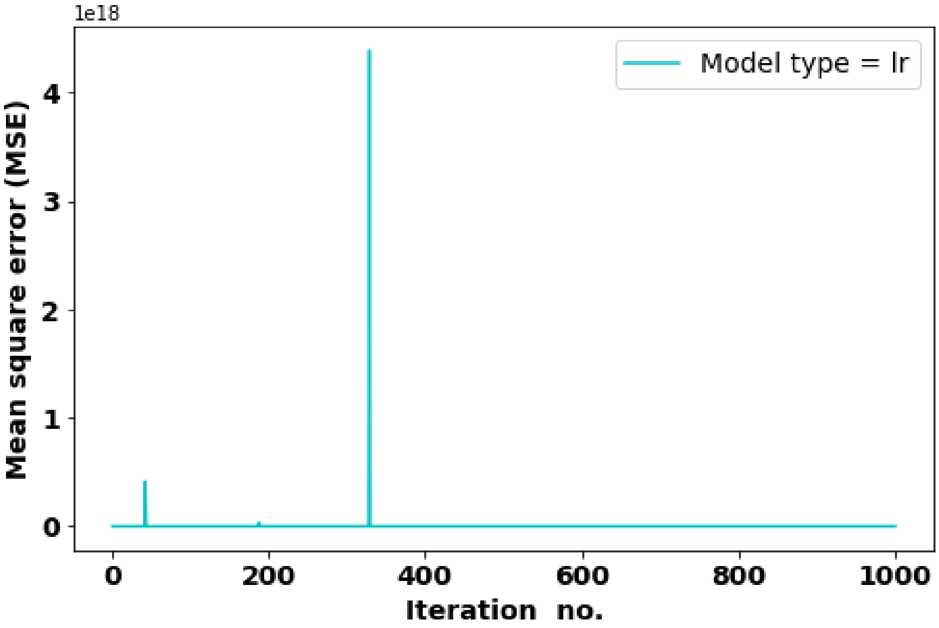


Figure 5. 6 Different iterations versus example mean square error (for test output y1), estimated through linear regression model.

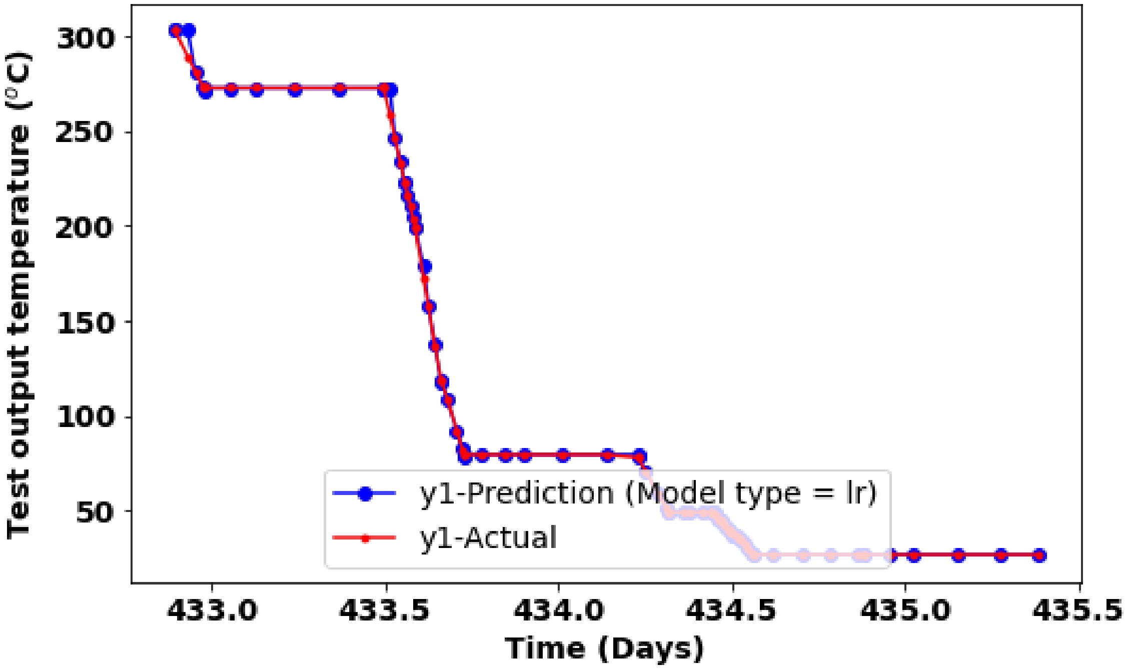


Figure 5. 7 Actual values (simulated through FE model) and their comparison with linear regression model predicted temperatures at nodal location 1 (i.e., output y1).

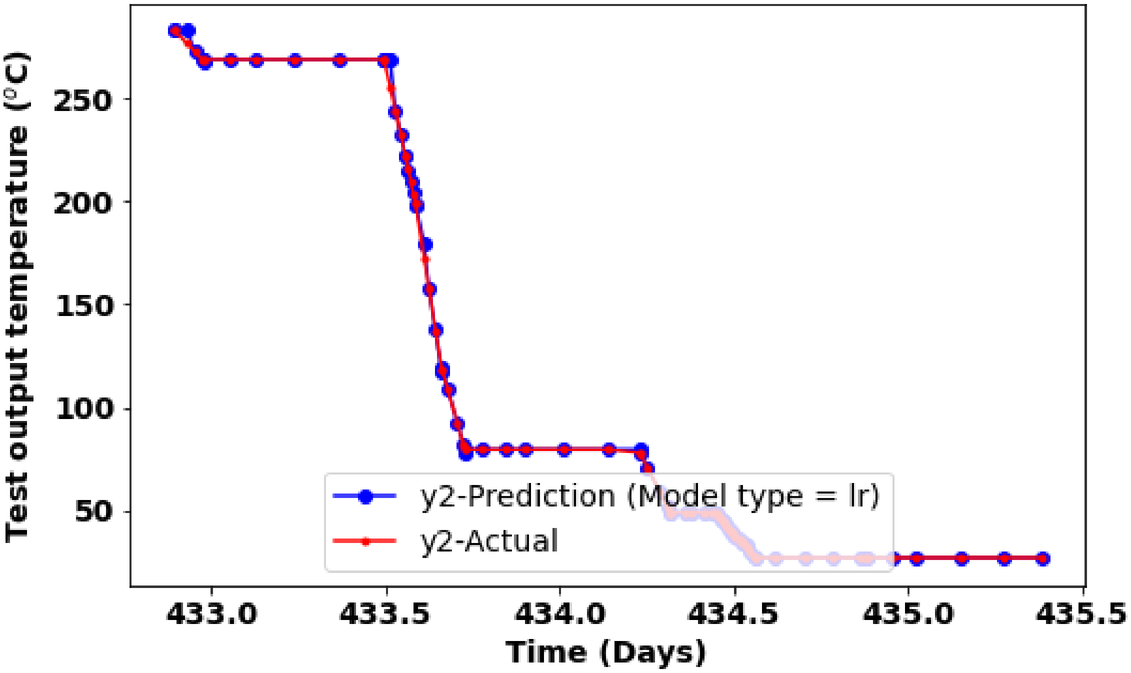


Figure 5. 8 Actual values (simulated through FE model) and their comparison with linear regression model predicted temperatures at nodal location 2 (i.e., output y2).

5.3 Quantitative Evaluation of Random-Forest (RF) Regression Model

Using the training input-output data given in section 5.1, and AI/ML based RF regression technique discussed in section 4.3 a MIMO model was inferred. The related model evaluation results are presented here. For example, Figure 5.9 shows the different iterations versus training scores R2, whereas the Figure 5.10 shows the corresponding iterations versus mean square error for test output y1). Figures 5.11 and 5.12 show the actual (simulated through FE model) versus RF regression model predicted temperatures at nodal location 1 (i.e., output y1 2 (i.e., output y2), respectively. All these figures show the RF model is far better compared to the linear model. Particularly looking at Figure 5.10 it can be seen that the maximum MSE is below 350 which is far less than the linear model.

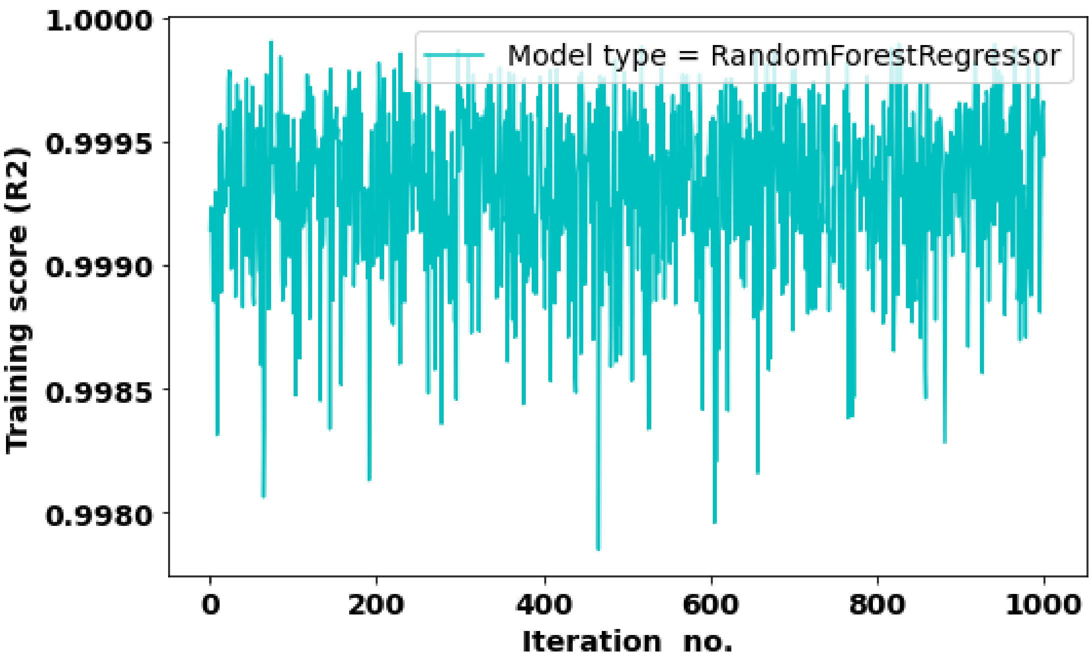


Figure 5. 9 Different iterations versus training scores R2 (i.e., the unform average of coefficients of determination for training outputs y1 and y2), estimated through Random-Forest regression model.

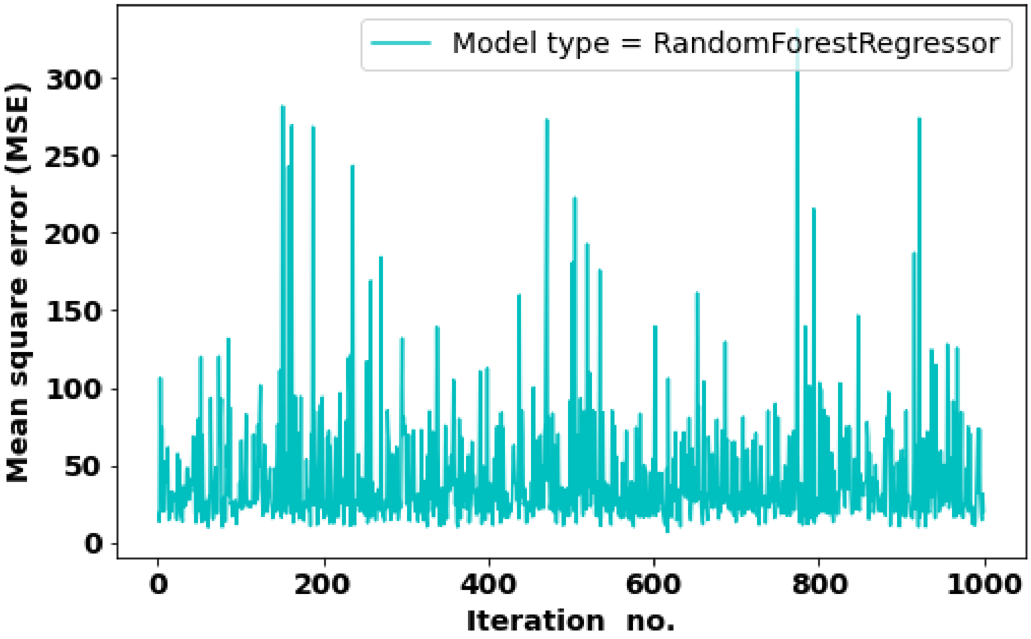


Figure 5. 10 Different iterations versus example mean square error (for test output y1), estimated through Random-Forest regression model.

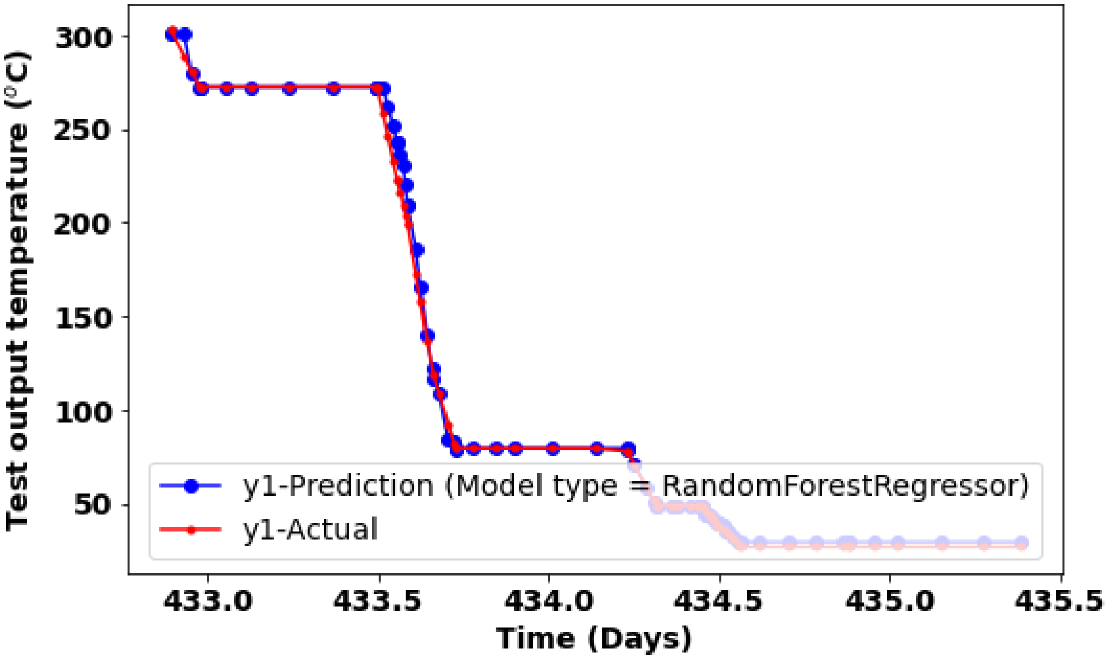


Figure 5. 11 Actual values (simulated through FE model) and their comparison with Random-Forest regression model predicted temperatures at nodal location 1 (i.e., output y1).

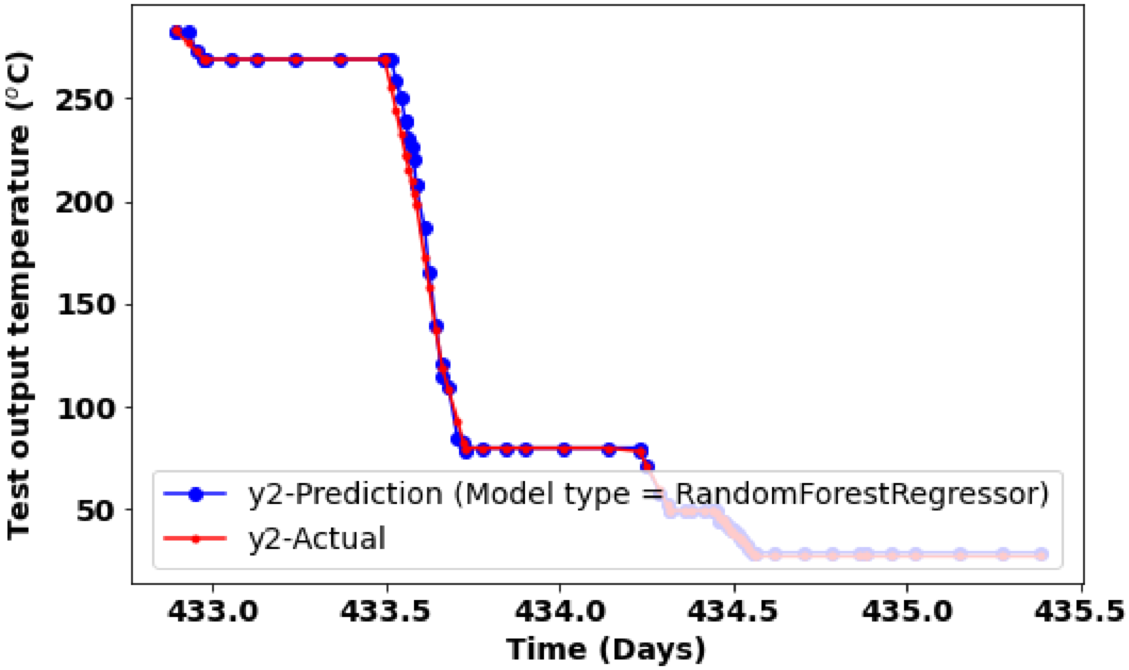


Figure 5. 12 Actual values (simulated through FE model) and their comparison with Random-Forest regression model predicted temperatures at nodal location 2 (i.e., output y2).

5.4 Quantitative Evaluation of Gradient-Boosting (GB) Regression Model

In addition to the AI/ML decision-tree based RF model, we also evaluated decision-tree based GB model (section 4.4). The model parameters were inferred using the training input-output data given in section 5.1. The related model evaluation results are presented here. Figures 5.13 and 5.14 respectively show the different iterations versus training scores R2 and MSE (for test output y1). Whereas, Figures 5.15 and 5.16 show the true/actual values (simulated through FE model) versus GB regression model predicted temperatures at nodal location 1 (i.e., output y1) and 2 (i.e., output y2), respectively. Comparing the GB model results with RF model results (e.g., comparing Figure 5.14 with 5.10), we can see the GB model perform very similar as RF model although with slightly higher MSE than the MSE of RF model.

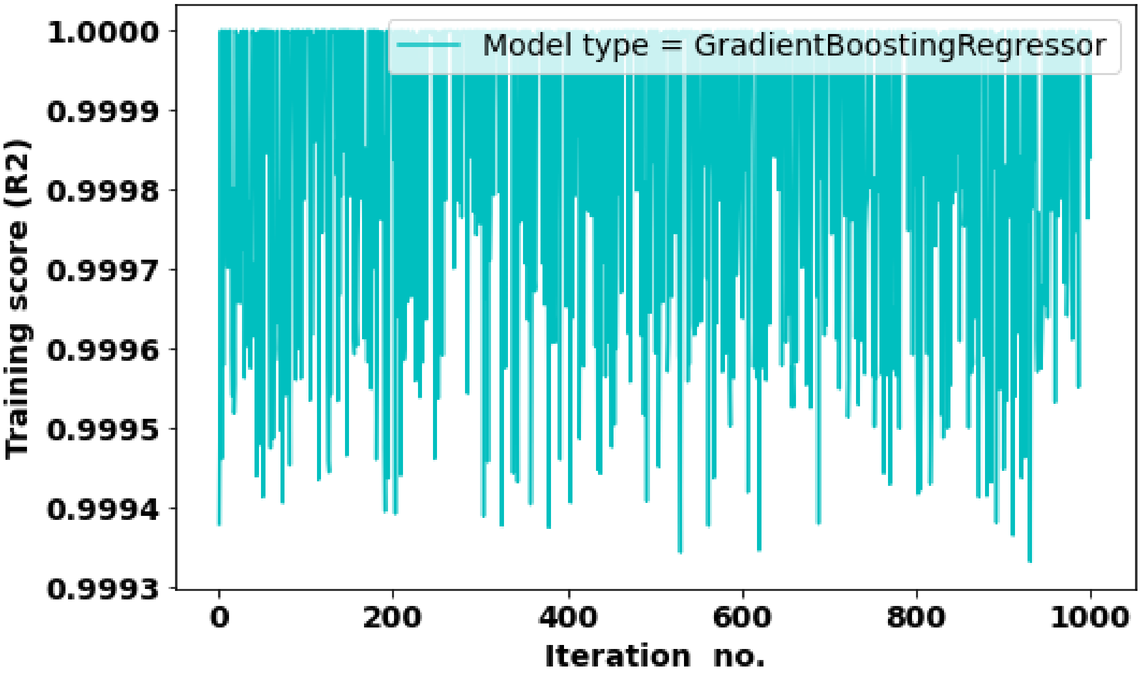


Figure 5. 13 Different iterations versus training scores R2 (i.e., the unform average of coefficients of determination for training outputs y1 and y2), estimated through Gradient-Boosting regression model.

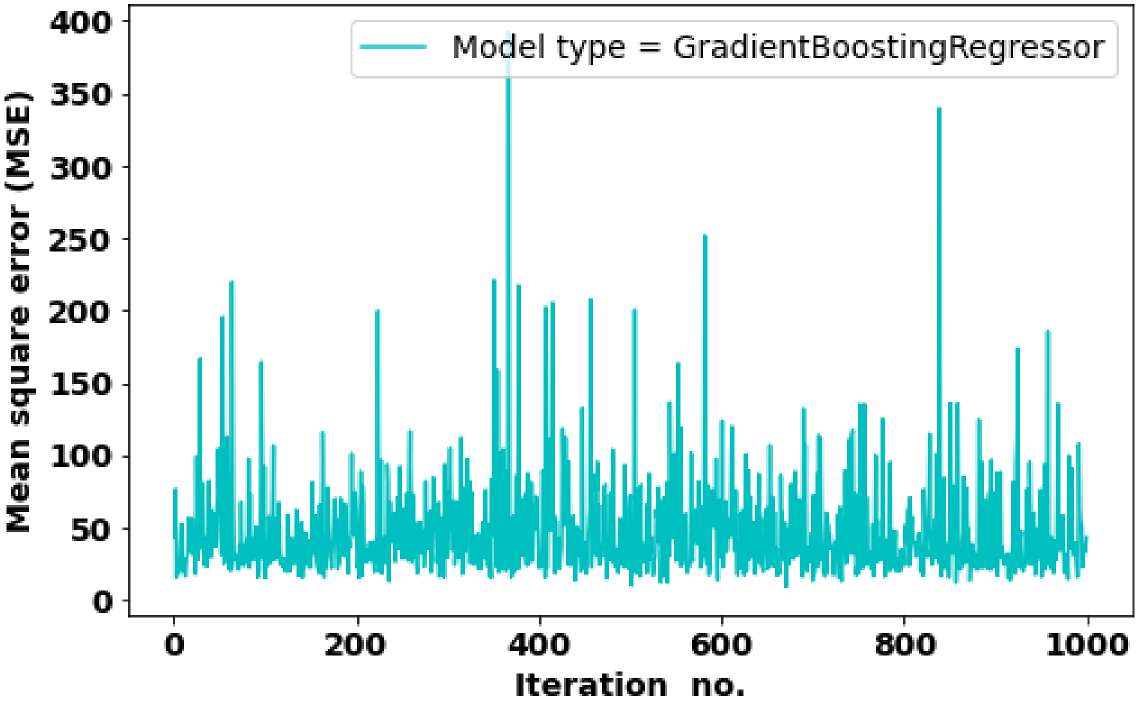


Figure 5. 14 Different iterations versus example mean square error (for test output y1), estimated through Gradient-Boosting regression model.

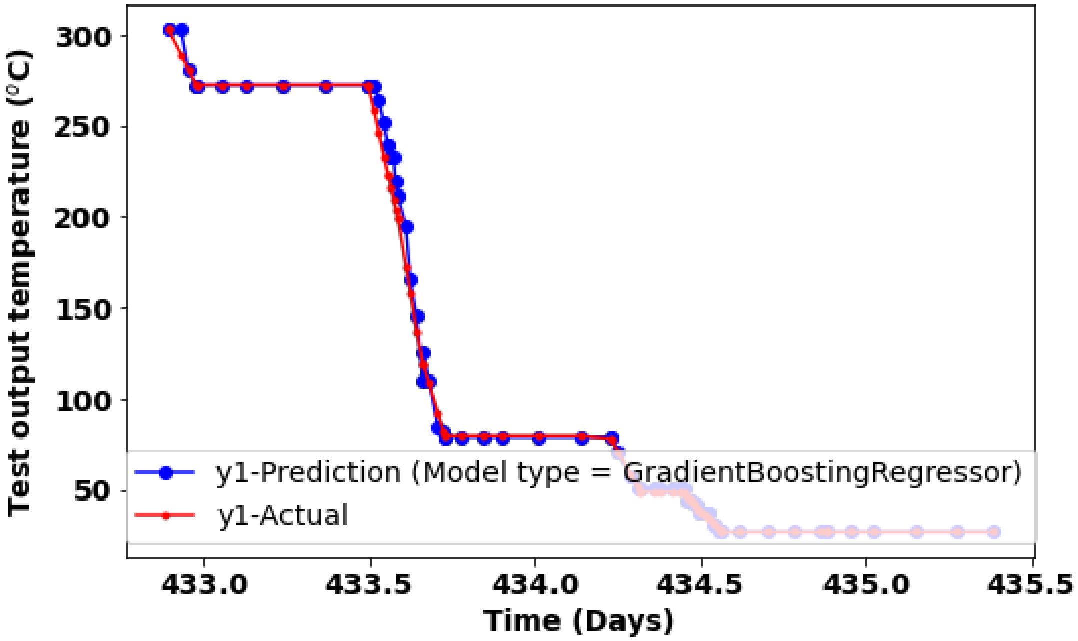


Figure 5. 15 Actual values (simulated through FE model) and their comparison with Gradient-Boosting regression model predicted temperatures at nodal location 1 (i.e., output y1).

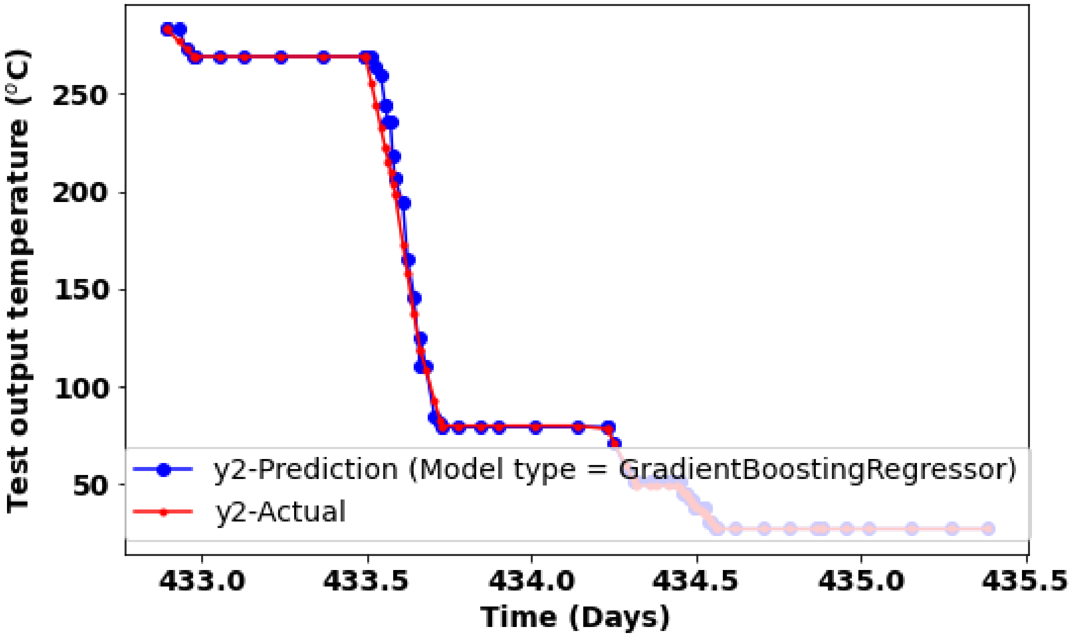


Figure 5. 16 Actual values (simulated through FE model) and their comparison with Gradient-Boosting regression model predicted temperatures at nodal location 2 (i.e., output y2).

5.5 Quantitative Evaluation of Support-Vector-Regression (SVM) Model with RBF Kernel

In addition to the linear and decision-tree based RF and GB models, we have also evaluated various Support-Vector-Regression (SVM) Models (section 4.5). This is using different kernel functions such as using the RBF, Sigmoid and linear kernel functions. In this section evaluation results with respect to RBF kernel function are presented. The related results for Sigmoid and linear kernel are presented in section 5.6 and 5.7, respectively. As before, the model parameters were inferred using the training input-output data given in section 5.1. Figures 5.17 and 5.18 show the iterations versus training scores R2 and MSE with respect to test output y1. Figures 5.19 and 5.20 show actual versus RBF-kernel-based SVM model predicted temperatures at nodal location 1 and 2 respectively. From Figures 5.17 to 5.20, the SVM model with RBF kernel poorly perform compared to the previously discussed linear, RF and GB models.

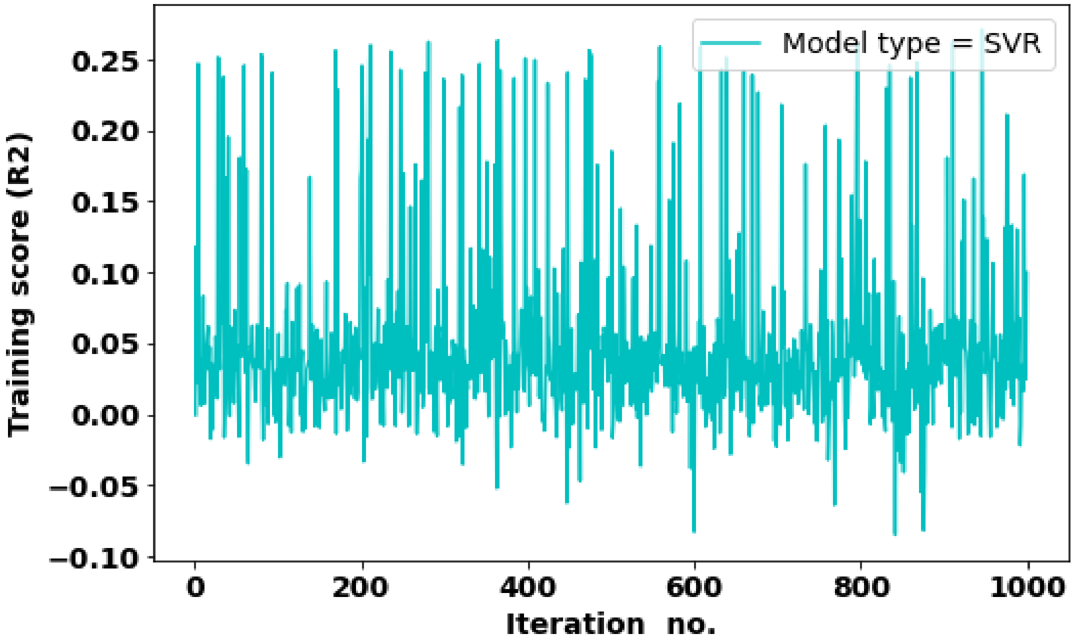


Figure 5. 17 Different iterations versus training scores R2 (i.e., the unform average of coefficients of determination for training outputs y1 and y2), estimated through Support-Vector-Machine regression model (with RBF kernel).

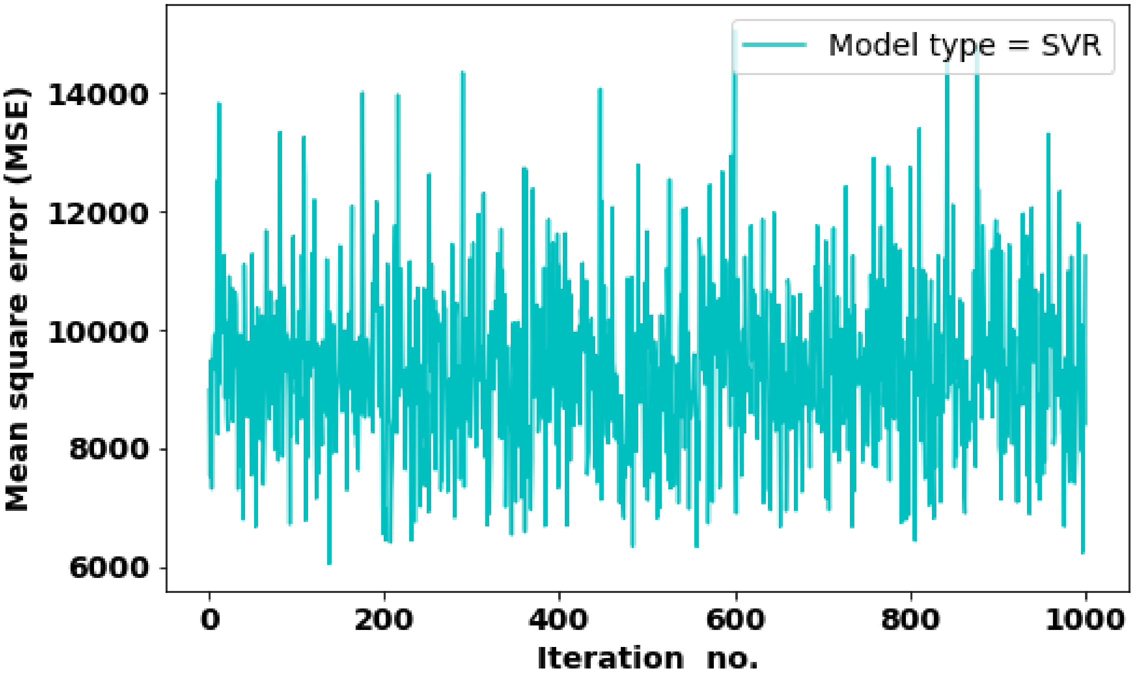


Figure 5. 18 Different iterations versus example mean square error (for test output y1), estimated through Support-Vector-Machine regression model (with RBF kernel).

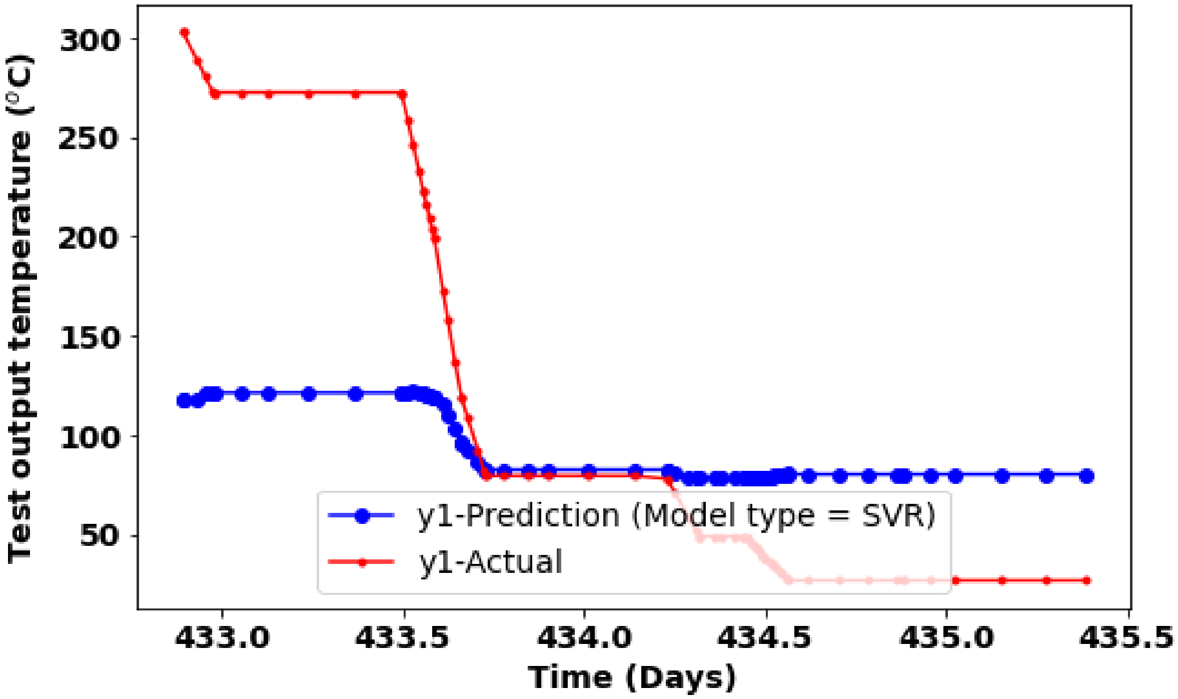


Figure 5. 19 Actual values (simulated through FE model) and their comparison with Support-Vector-Machine model (with RBF kernel) predicted temperatures at nodal location 1 (i.e., output y1).

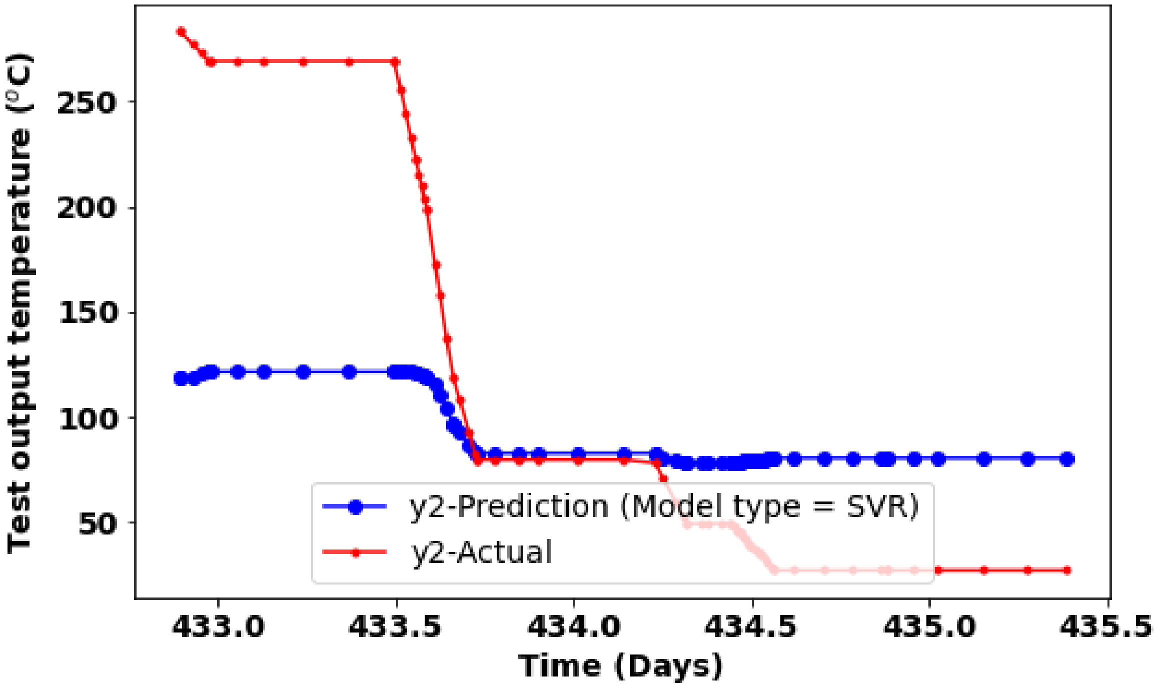


Figure 5. 20 Actual values (simulated through FE model) and their comparison with Support-Vector-Machine model (with RBF kernel) predicted temperatures at nodal location 2 (i.e., output y2).

5.6 Quantitative Evaluation of Support-Vector-Regression-Model with Sigmoid Kernel Function

In this section evaluation results with respect to Sigmoid-kernel-function based SVM model are presented. Figures 5.21 to 5.24 show the related model evaluation results. From these figures, similar as RBF-kernel-based SVM model, the Sigmoid-kernel-based SVM model performs poorly compared to the previously discussed linear, RF and GB models.

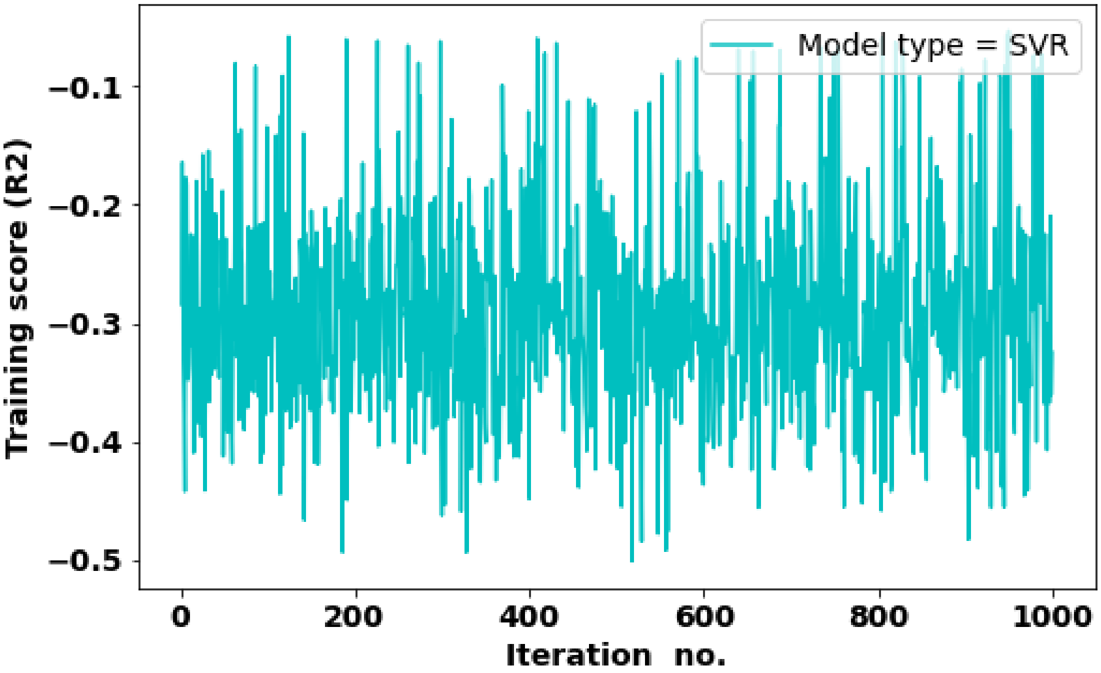


Figure 5. 21 Different iterations versus training scores R2 (i.e., the unform average of coefficients of determination for training outputs y1 and y2), estimated through Support-Vector-Machine regression model (with Sigmoid kernel).

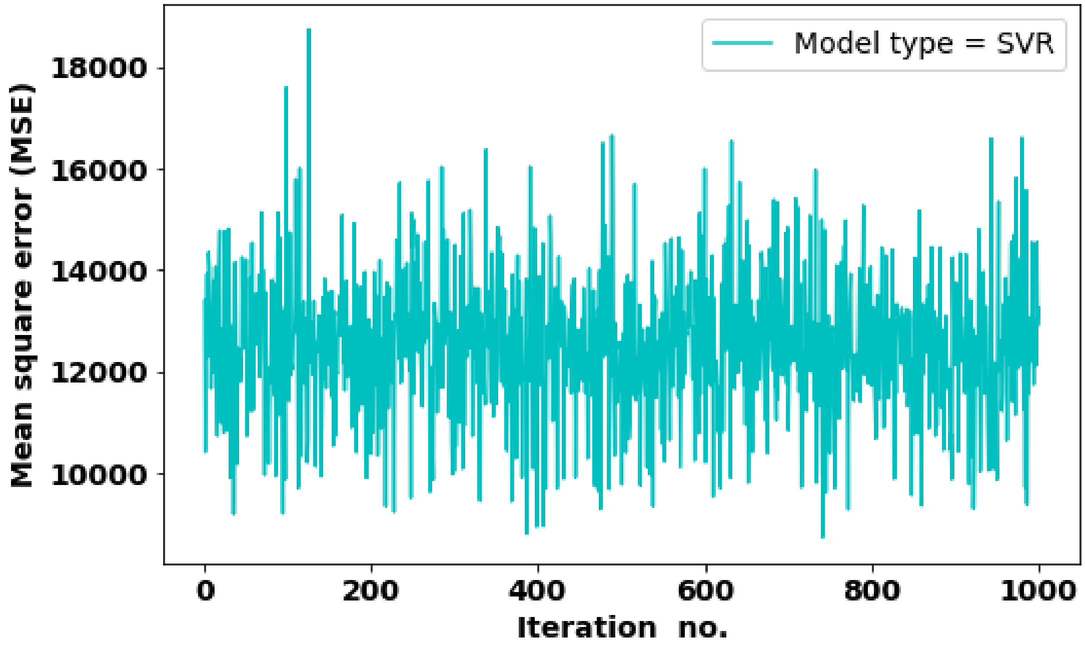


Figure 5. 22 Different iterations versus example mean square error (for test output y1), estimated through Support-Vector-Machine regression model (with Sigmoid kernel).

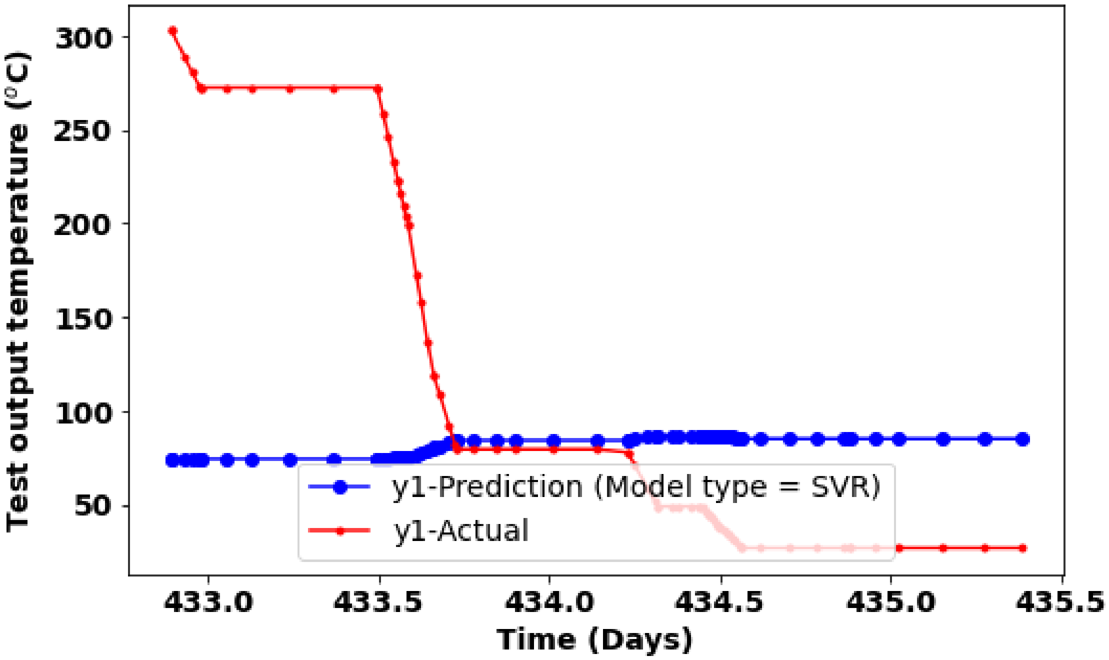


Figure 5. 23 Actual values (simulated through FE model) and their comparison with Support-Vector-Machine model (with Sigmoid kernel) predicted temperatures at nodal location 1 (i.e., output y1).

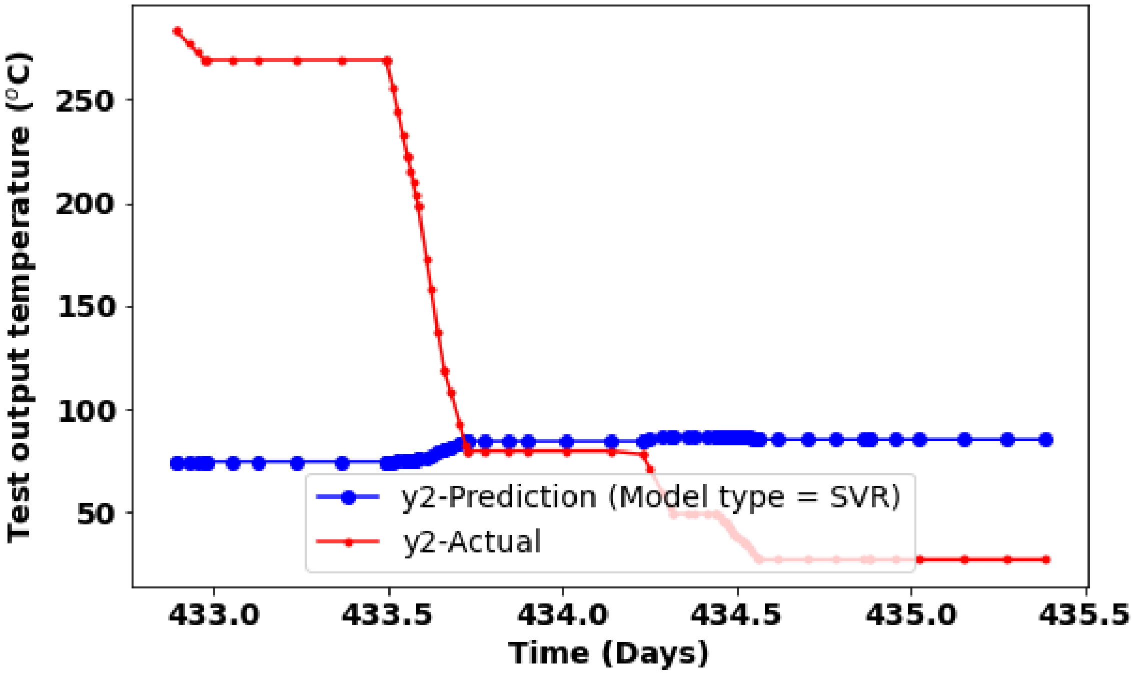


Figure 5. 24 Actual values (simulated using FE model) and their comparison with Support-Vector-Machine model (with Sigmoid kernel) predicted temperatures at nodal location 2 (i.e., output y2).

5.7 Quantitative Evaluation of Support-Vector-Regression-Model with Linear Kernel Function

In this section evaluation results with respect to linear-kernel-function-based SVM model are presented. Figures 5.25 to 5.28 show the related model evaluation results. From these figures, the linear-kernel-based SVM model not only outperform the RBF and Sigmoid kernel-based SVM models, but also predict far better compared to the previously discussed multi-variate linear, RF and GB models. For example, the maximum estimated MSE for linear-kernel-based SVM model is 55, whereas the corresponding values for multi-variate linear, RF and GB models are 1e18, 350 and 400, respectively. These types of result would guide to select appropriate AI/ML models. Nevertheless, the multivariate linear, RF, GB and only the linear-kernel-based SVM models are further evaluated for spatial-temporal temperature predictions at thousands of point-clouds, which results follow.

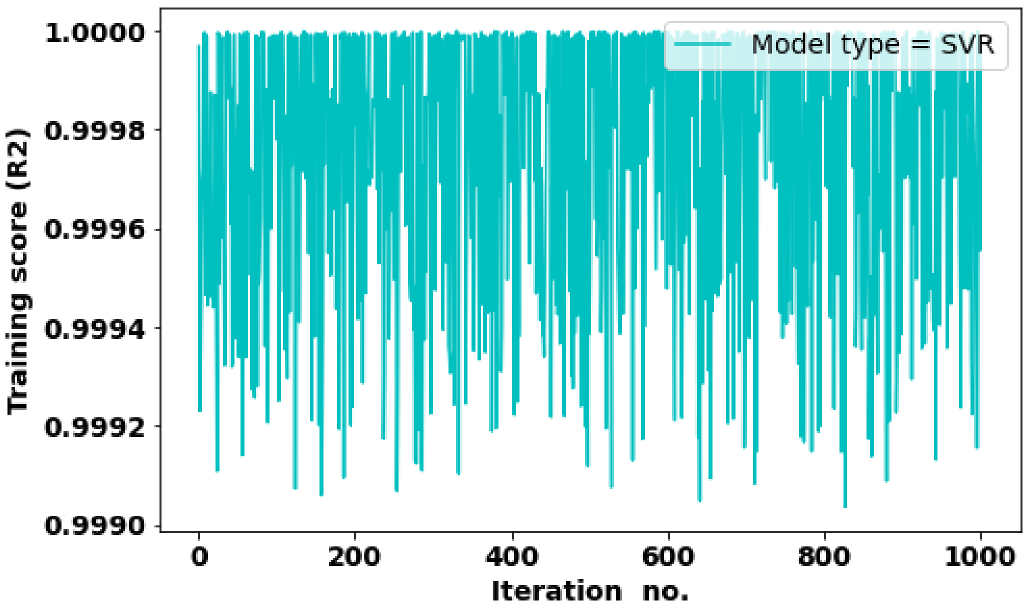


Figure 5. 25 Different iterations versus training scores R2 (i.e., the unform average of coefficients of determination for training outputs y1 and y2), estimated through Support-Vector-Machine regression model (with linear kernel).

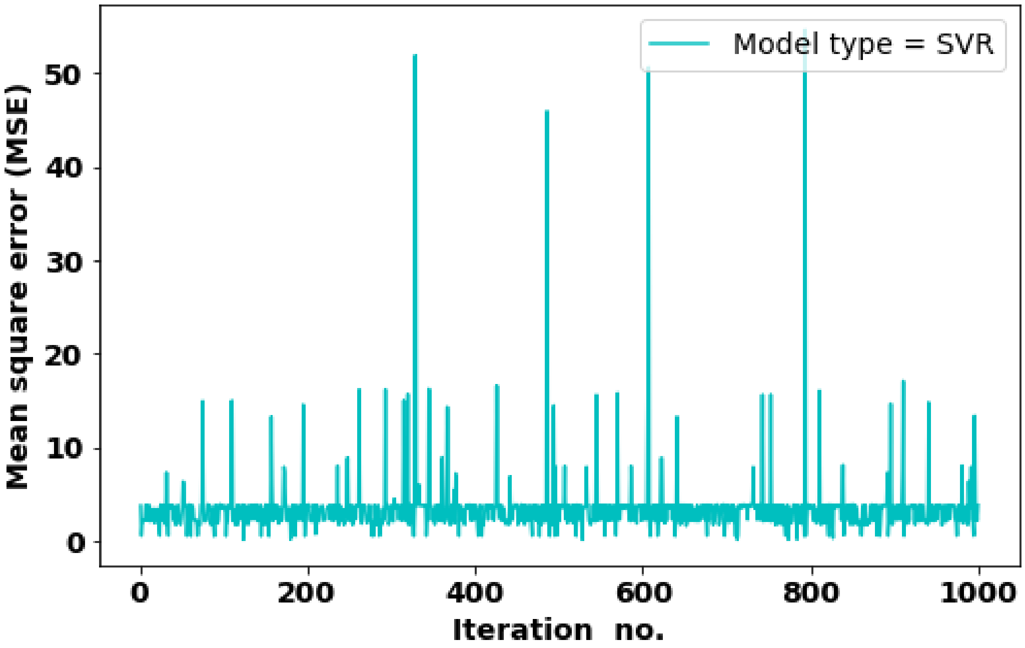


Figure 5. 26 Different iterations versus example mean square error (for test output y1), estimated through Support-Vector-Machine regression model (with linear kernel).

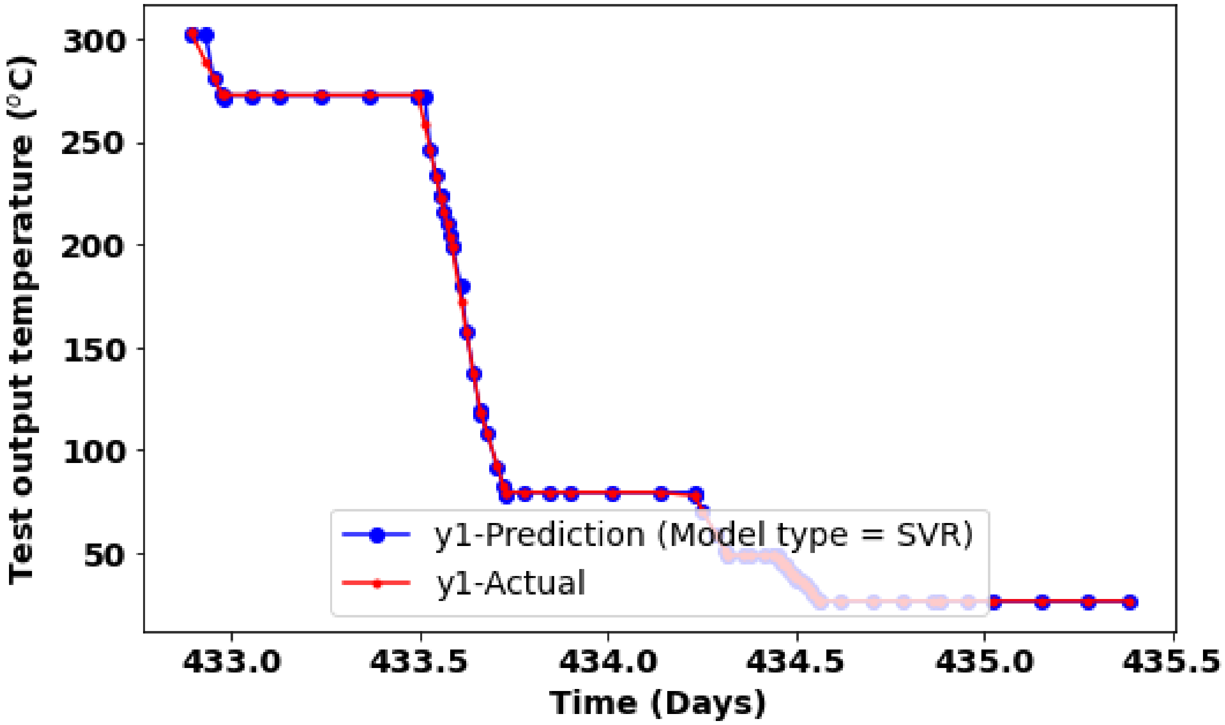


Figure 5. 27 Actual values (simulated through FE model) and their comparison with Support-Vector-Machine model (with linear kernel) predicted temperatures at nodal location 1 (i.e., output y1).

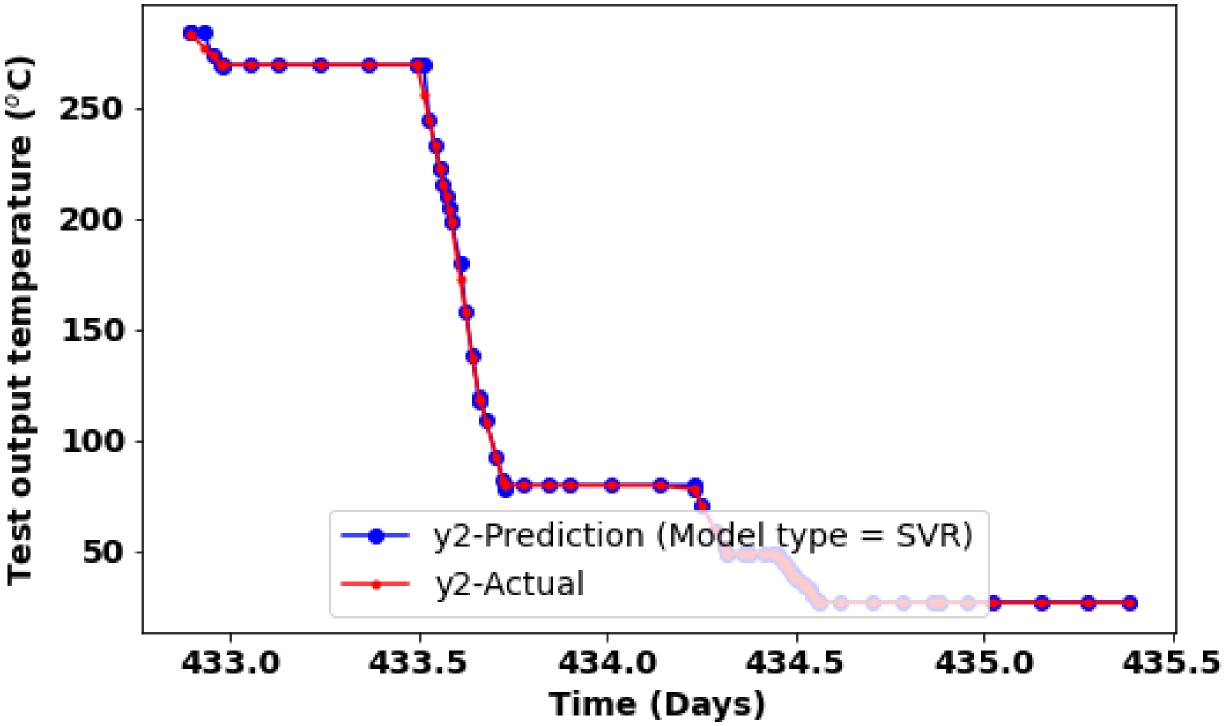


Figure 5. 28 Actual values (simulated through FE model) and their comparison with Support-Vector-Machine model (with linear kernel) predicted temperatures at nodal location 2 (i.e., output y2).

6 Multi-Time-Series Temperature Prediction in a SG-Tube: Predictions at Thousands of 3D Point-clouds Under a Design-Basis Loading Cycle

In section 5, different AI/ML models were quantitatively evaluated with respect to only two time-series at two random locations. However, a 3D component can have significant temperature variations along all the three dimensions. Differential temperature in a component can lead to significant thermal strain induced stress and associated accelerated damage progression. Unless the 3D variations of the temperature along with their time-dependency accurately predicted, the stress-strain and the associated damage state cannot be accurately predicted. In this section and in section 7, we present some of the example results related to 3D prediction of the time-series temperature at thousands of locations such as at all the 3720 FE nodes of the U-bend tube shown in Figure 2.15. The prediction results with respect to design-basis (DB) loading cycle are presented in this section, whereas the corresponding results with respect to grid-load-following (GLF) loading cycle are presented in section 7. The results are presented for multivariate linear, decision-tree based RF and GB models and for the linear-kernel-function based SVM model. Below presents the details of the training data and the model evaluation results.

6.1 Training Data Used for Design-Basis Loading Case

Before any prediction, the model parameters with respect to different AI/ML models need to be inferred based on training or historical data. We used the same process inputs (refer Figure 5.2) as discussed in section 5. However, unlike the models in section 5 (which were trained for 2 outputs) the models discussed in section 6 and 7 were trained for 3720 geometry-dependent outputs. This leads to a MIMO model with 4 inputs and 3720 outputs. Similar as discussed in section 5, we only considered the heat-up portion of the DB cycle for training the model. The corresponding FE-model simulated temperatures at 3720 geometry-dependent locations (Figure 2.15) were used as training outputs. Figure 6.1 shows the FE-model simulated temperature data those were used as training outputs.

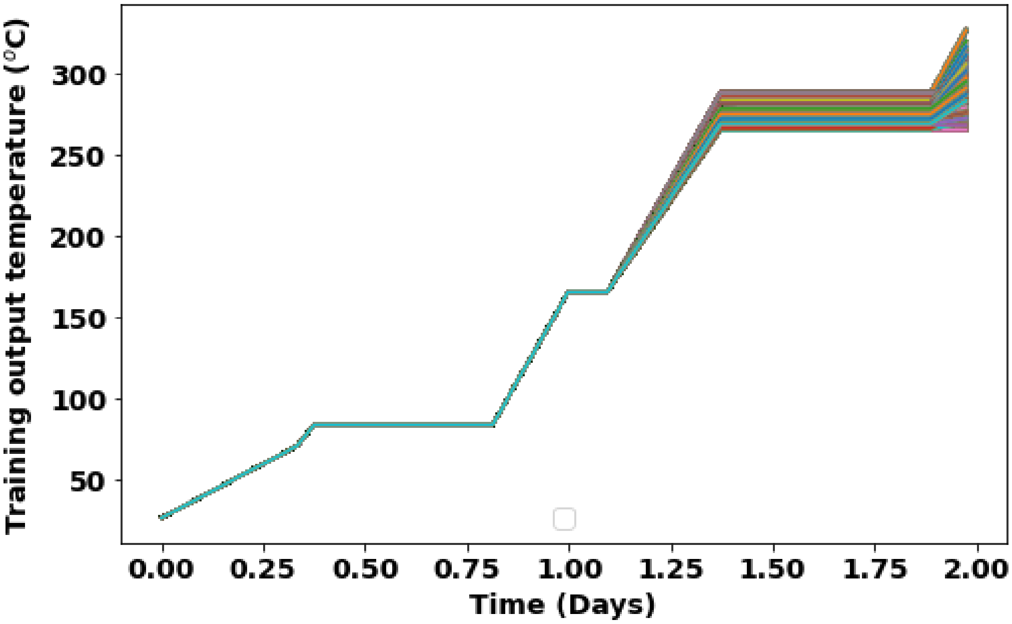


Figure 6. 1 FE-model simulated training-output-temperature data at 3720 spatial locations (covering only during the heat-up operation of a design-basis type loading cycle), which were used for training all the discussed AI/ML models which results are presented in section 6.

6.2 Multivariate Linear-Regression (LR) Model Based Temperature Prediction at Thousands of Locations under a Design-Basis Loading Cycle

The evaluation results related to the multivariate linear regression model are presented here. Using the trained model (discussed above), the spatial-temporal temperatures were predicted for the process inputs given in Figure 5.3 (which are same as for all the results discussed in sections 5 and 6). Figure 6.2 shows different output no. versus corresponding training scores R2. The R2 scores are the coefficients of determination for the individual training outputs at nodal locations 1 to 3720. Figure 6.3 shows the output no. versus mean square error for the individual test outputs at all nodal locations 1 to 3720 (refer Figure 2.15). Note unlike the training outputs (refer Figure 6.1, which covers the heat-up regime of DB cycle), the test output time-series covers totally different regime of the DB cycle such as portion of the steady-state power operation regime and the entire cool-down regime of the DB loading cycle (refer Figure 5.3 for corresponding test inputs). Figure 6.4 shows the AI/ML (linear regression) model predicted temperatures (or test outputs) at nodal locations 1 to 3720. Whereas Figure 6.5 shows the corresponding actual values or the FE model simulated temperatures. From Figures 6.4 and 6.5 it can be seen that there is a good qualitative match between AI/ML and FE model predicted temperature time series. However, looking at Figure 6.3, MSE are very high for some of the outputs. This observation is similar as the MSE results (for multivariate linear model) discussed in section 5. These types of results further conform that, although multivariate linear models are mathematically elegant, are not robust for the mentioned predictive model applications.

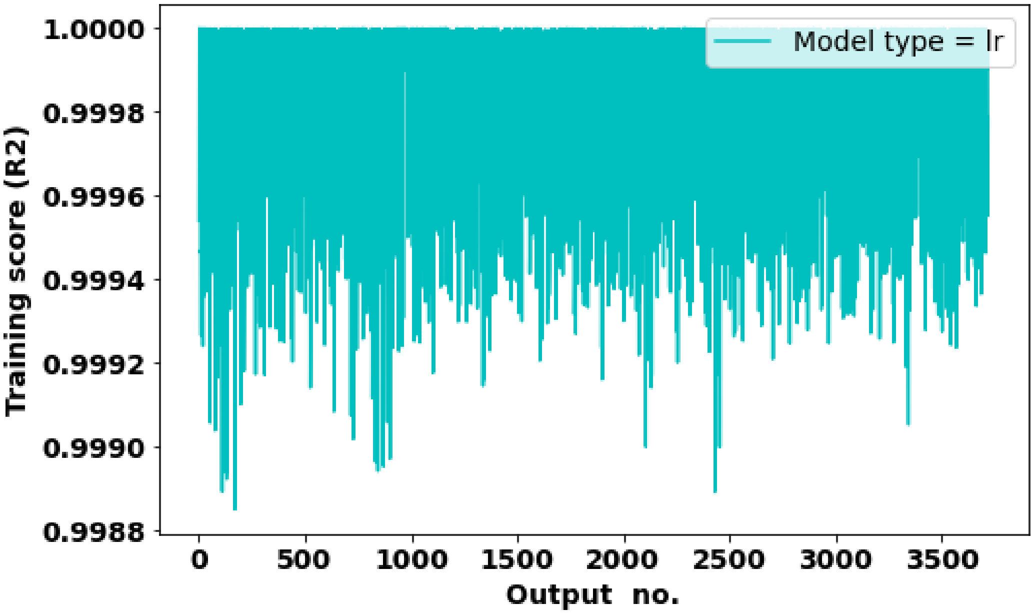


Figure 6. 2 Different output no. versus corresponding training scores R2 (i.e., the individual coefficients of determination for training outputs at nodal locations 1 to 3720), estimated through AI/ML (linear regression) model.

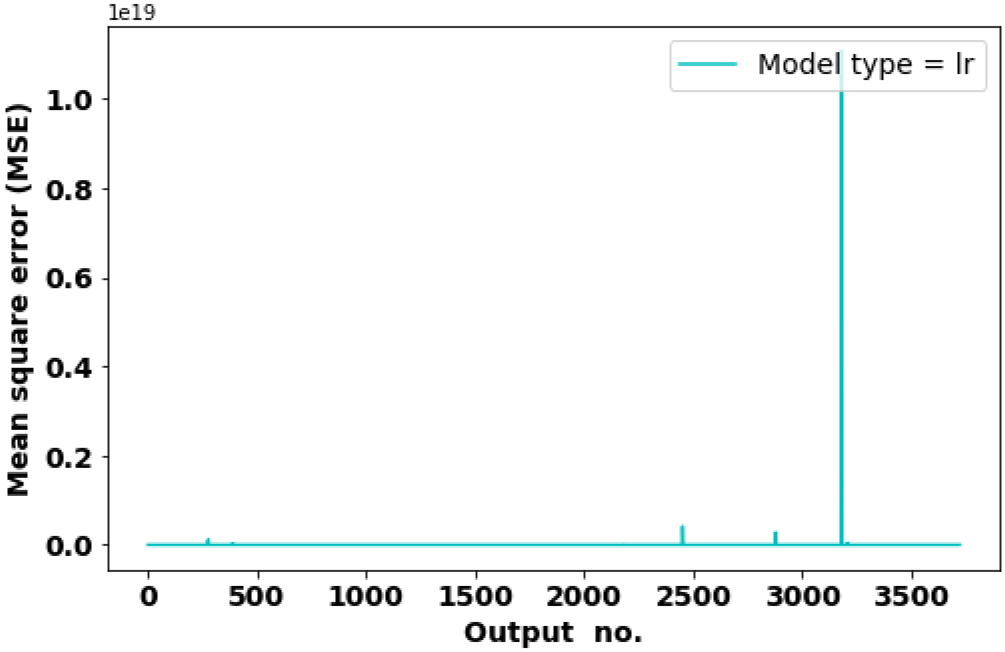


Figure 6. 3 Different output no. versus corresponding mean square error (for test outputs at nodal locations 1 to 3720), estimated through AI/ML (linear regression) model. Each test output time-series covers a portion of the steady-state power operation regime and the entire cool-down regime of a design-basis type loading cycle.

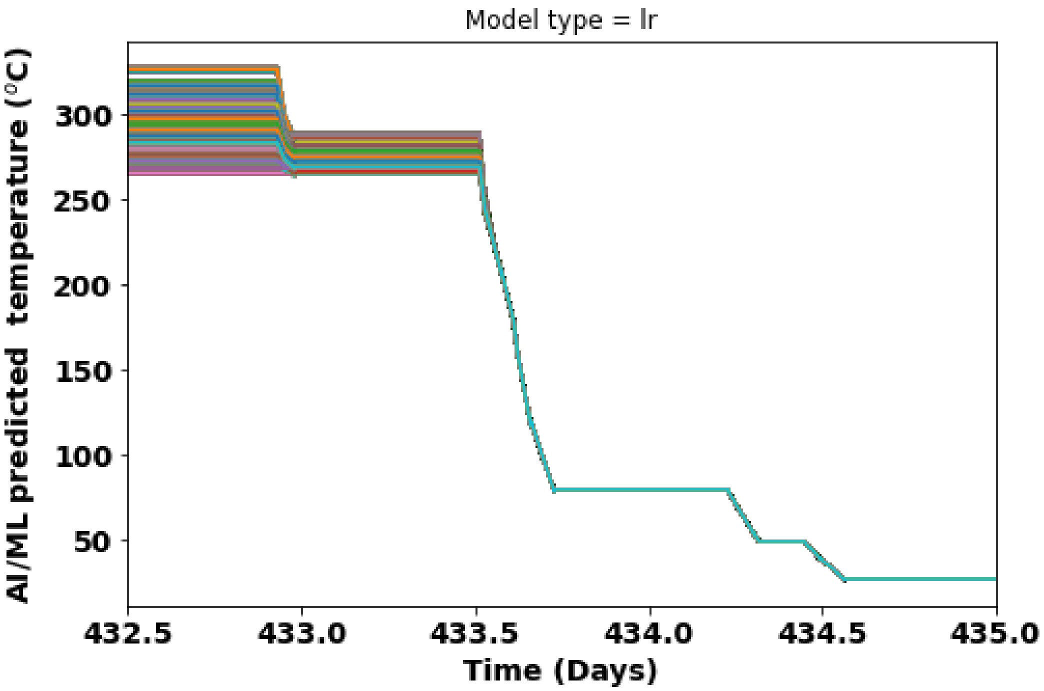


Figure 6. 4 AI/ML (linear regression) model predicted temperatures (or test outputs) at nodal locations 1 to 3720. Each test output time-series covers a portion of the steady-state power operation regime and the entire cool-down regime of a design-basis type loading cycle.

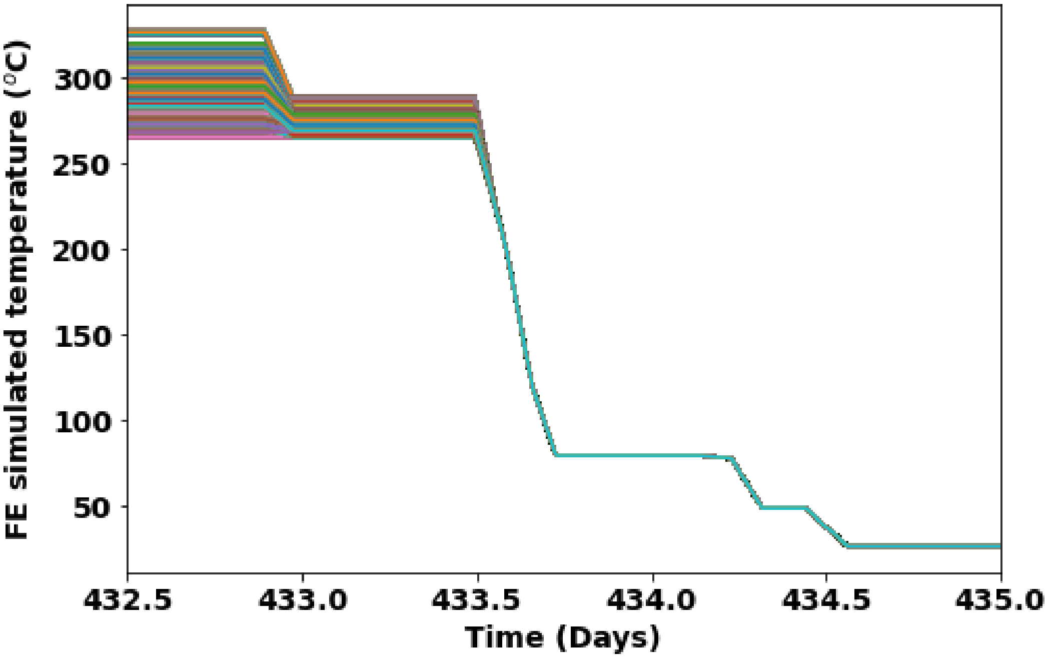


Figure 6. 5 Actual FE model simulated temperatures (or test outputs) at nodal locations 1 to 3720. Each test output time-series covers a portion of the steady-state power operation regime and the entire cool-down regime of a design-basis type loading cycle.

6.3 Random-Forest (RF) Regression Model Based Temperature Prediction at Thousands of Locations under Design-Basis Loading Cycle

The evaluation results related to Random-Forest (RF) regression model-based predictions at thousands of 3D locations are presented here. For example, Figures 6.6 shows the different training output no. versus training scores R2, whereas Figure 6.7 show the different test output no. versus MSE. Figure 6.8 shows the RF model predicted temperatures (or test outputs) at all nodal locations 1 to 3720 (refer Figure 2.15). Comparing Figure 6.8 with corresponding FE predicted temperatures (Figure 6.5), it can be seen there is a good qualitative comparison between RF model predicted temperature and the corresponding FE simulated temperature. These types of results give the confidence that without a FE model, the spatial-temporal temperature at 3D locations can be predicted in real time for a given process inputs (measure in real time). From the Figure 6.7 the maximum MSE is approximately 550 which is very similar to the results discussed in section 5.

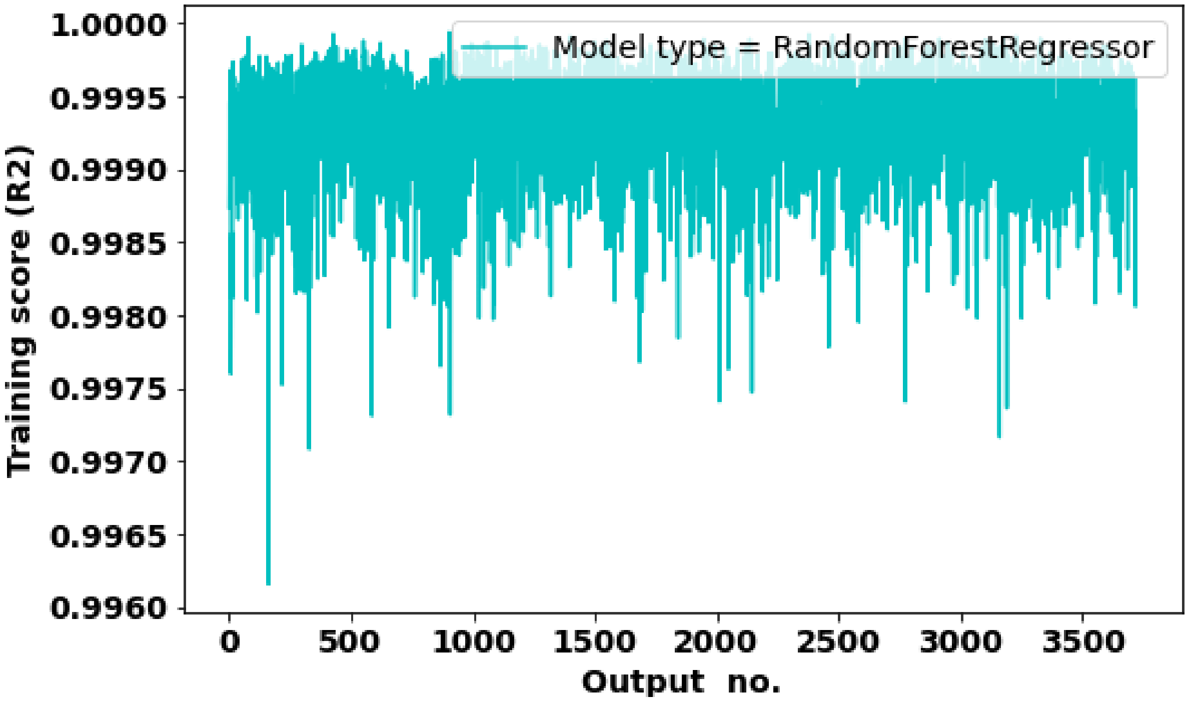


Figure 6. 6 Different output no. versus corresponding training scores R2 (i.e., the individual coefficients of determination for training outputs at nodal locations 1 to 3720), estimated through AI/ML (Random-Forest regression) model. Each test output time-series covers a portion of the steady-state power operation regime and the entire cool-down regime of a design-basis type loading cycle.

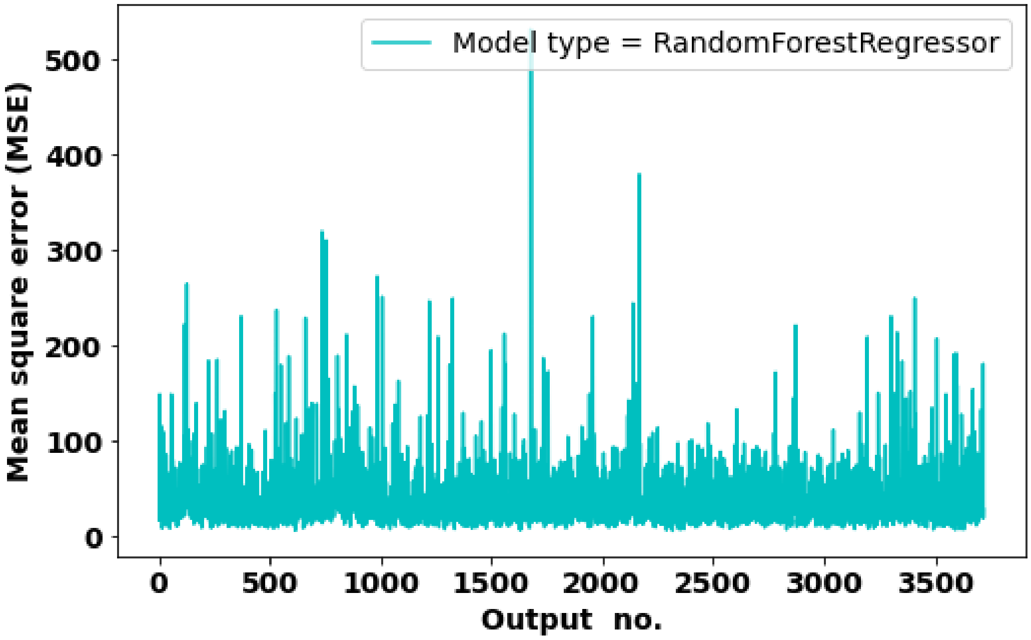


Figure 6. 7 Different output no. versus corresponding mean square error (for test outputs at nodal locations 1 to 3720), estimated through AI/ML (Random-Forest regression) model. Each test output time-series covers a portion of the steady-state power operation regime and the entire cool-down regime of a design-basis type loading cycle.

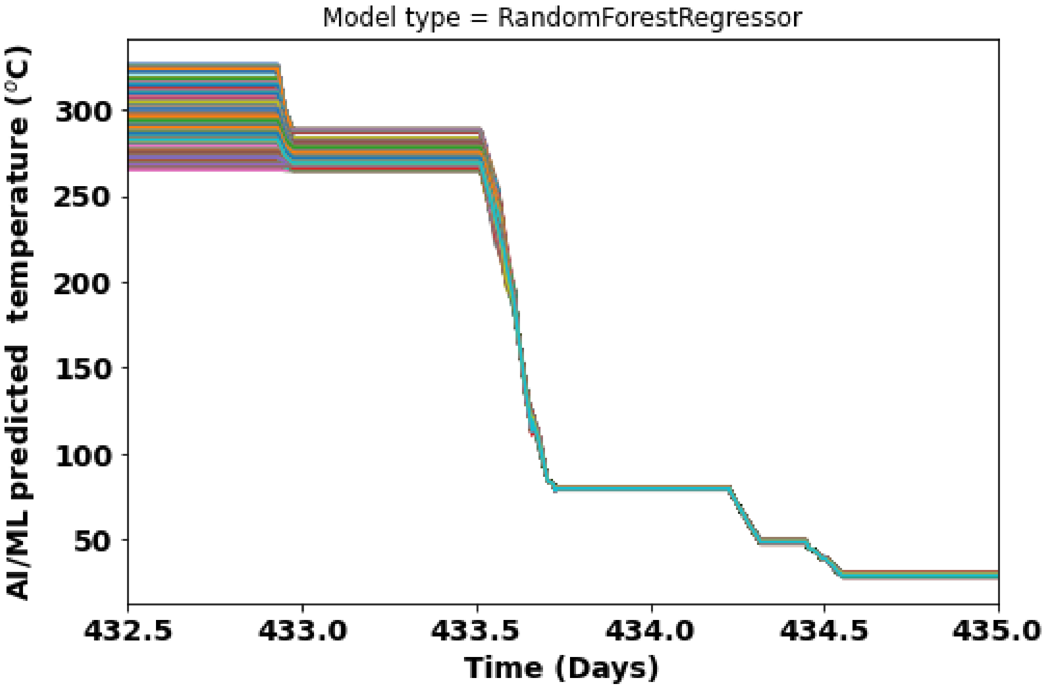


Figure 6. 8 AI/ML (Random-Forest regression) model predicted temperatures (or test outputs) at nodal locations 1 to 3720. Each test output time-series covers a portion of the steady-state power operation regime and the entire cool-down regime of a design-basis type loading cycle.

6.4 Gradient-Boosting (GB) Regression Model Based Temperature Prediction at Thousands of Locations under Design-Basis Loading Cycle

The evaluation results related to Gradient-Boosting (GB) regression model-based predictions at thousands of 3D locations are presented here. Figure 6.9 shows the related training output no. versus training scores R2. Whereas Figure 6.10 shows the related test output no. versus MSE of test outputs at nodal locations 1 to 3720. Figure 6.11 shows the GB model predicted temperatures (or test outputs) at all the nodal locations 1 to 3720 (refer Figure 2.15). Similar as RF model, the GM model shows a good qualitative comparison between GB model predicted temperature and the corresponding FE simulated temperature (Figure 6.5). In addition, the GB model produce a maximum MSE value of 520, which is very similar to the maximum MSE value obtained for corresponding RF model. These results implies that the accuracy for RF and GB model are very similar.

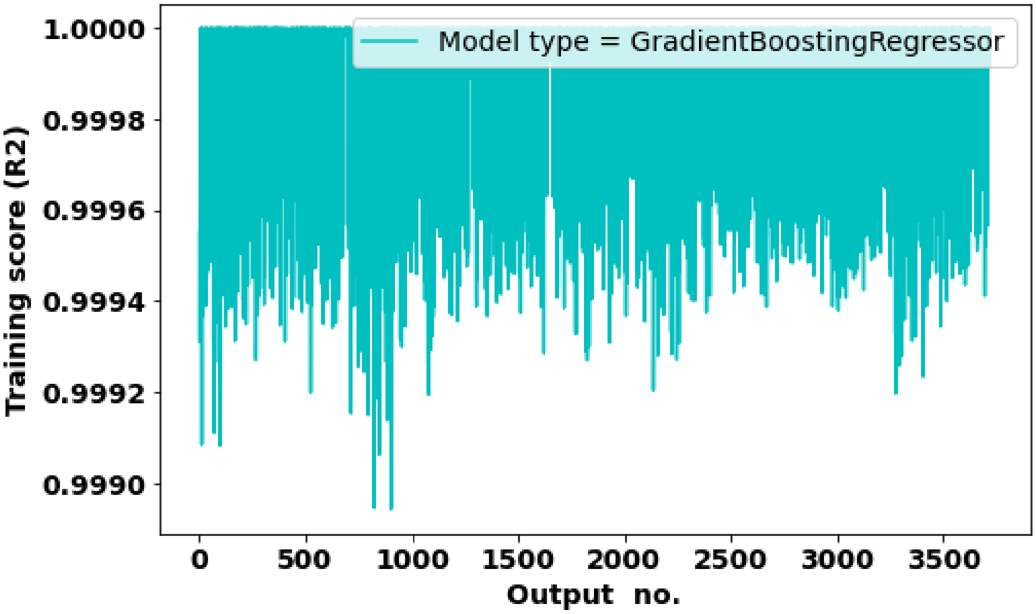


Figure 6. 9 Different output no. versus corresponding training scores R2 (i.e., the individual coefficients of determination for training outputs at nodal locations 1 to 3720), estimated through AI/ML (Gradient-Boosting regression) model. Each test output time-series covers a portion of the steady-state power operation regime and the entire cool-down regime of a design-basis type loading cycle.

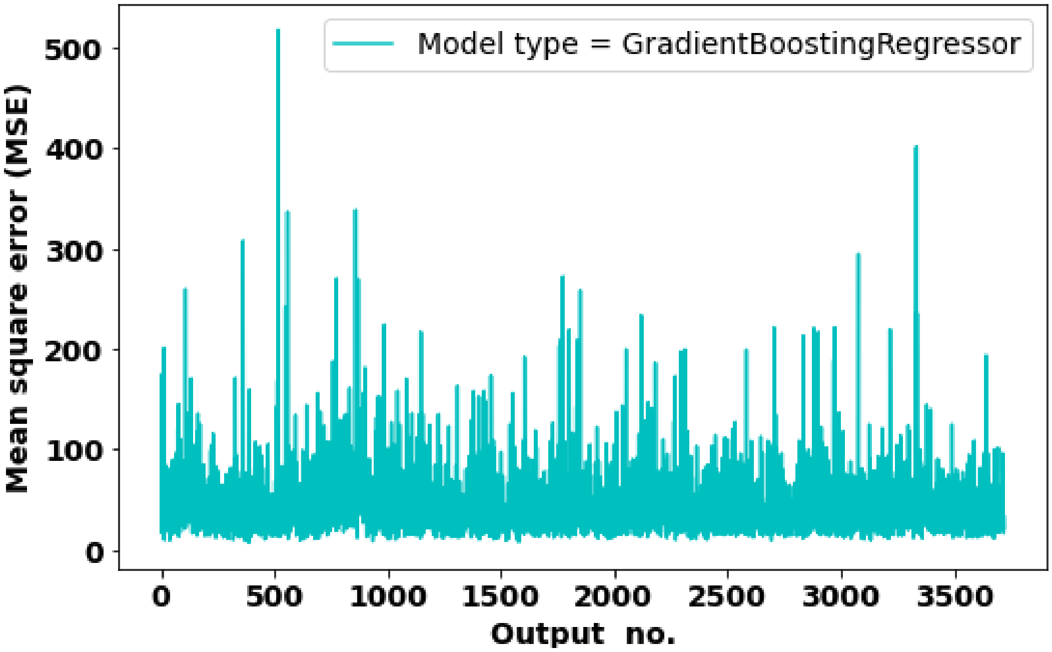


Figure 6. 10 Different output no. versus corresponding mean square error (for test outputs at nodal locations 1 to 3720), estimated through AI/ML (Gradient-Boosting regression) model. Each test output time-series covers a portion of the steady-state power operation regime and the entire cool-down regime of a design-basis type loading cycle.

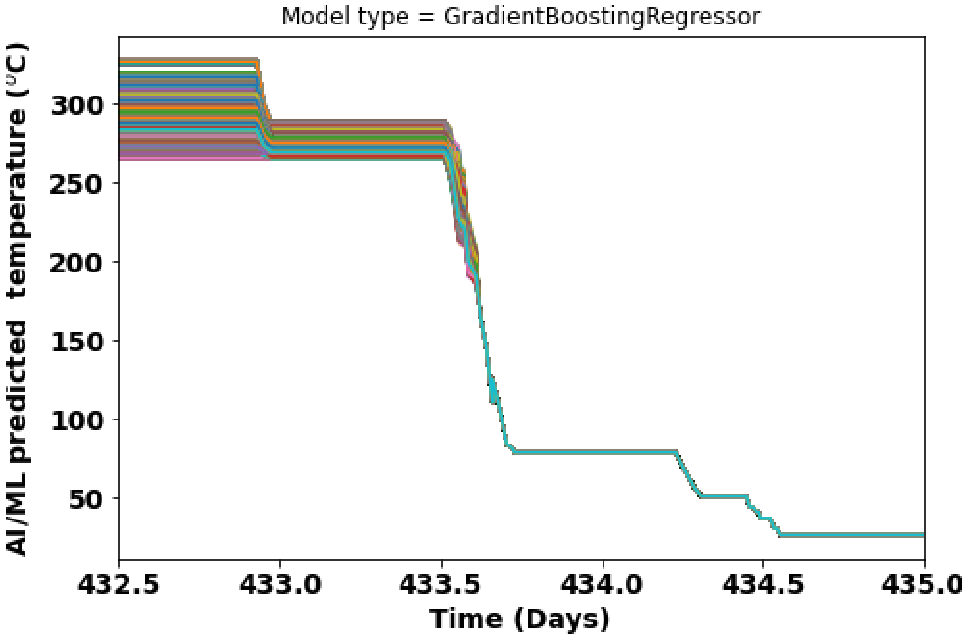


Figure 6. 11 AI/ML (Gradient-Boosting regression) model predicted temperatures (or test outputs) at nodal locations 1 to 3720. Each test output time-series covers a portion of the steady-state power operation regime and the entire cool-down regime of a design-basis type loading cycle.

6.5 Support-Vector-Machine (SVM) Regression Model Based Temperature Prediction at Thousands of Locations under Design-Basis Loading Cycle

The results related to SVM-model based predictions at thousands of 3D locations are presented here. The SVM model is based on linear kernel function. Figures 6.12 and 6.13 respectively show the training output no. versus training scores R2 and the test output no. versus MSE of test outputs at nodal locations 1 to 3720. Figure 6.14 shows the SVM model predicted temperatures at nodal locations 1 to 3720 (refer Figure 2.15). Comparing Figure 6.14 with Figure 6.5, similar as RF and GM models, the SVM model shows a good qualitative comparison with FE simulated temperatures (Figure 6.5). However, comparing the MSE plots (Figure 6.7 for RF, Figure 6.10 for GB and Figure 6.13 for SVM models) it can be seen that the SVM model is among the best with lowest MSE values among all the models. The maximum observed MSE is approximately 65 (considering all the 3720 temperature prediction locations).

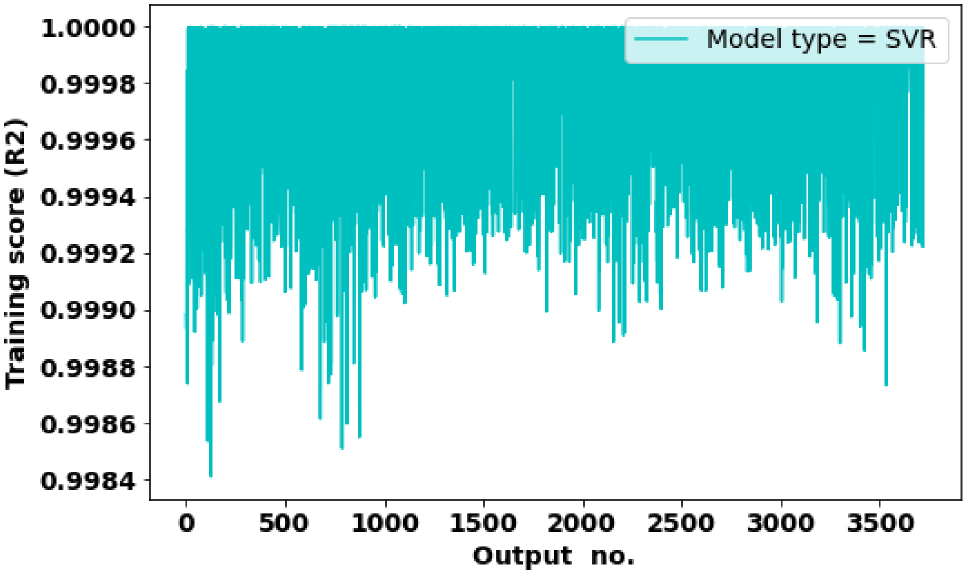


Figure 6. 12 Different output no. versus corresponding training scores R2 (i.e., the individual coefficients of determination for training outputs at nodal locations 1 to 3720), estimated through AI/ML (Support-Vector-Machine regression with linear kernel function) model. Each test output time-series covers a portion of the steady-state power operation regime and the entire cool-down regime of a design-basis type loading cycle.

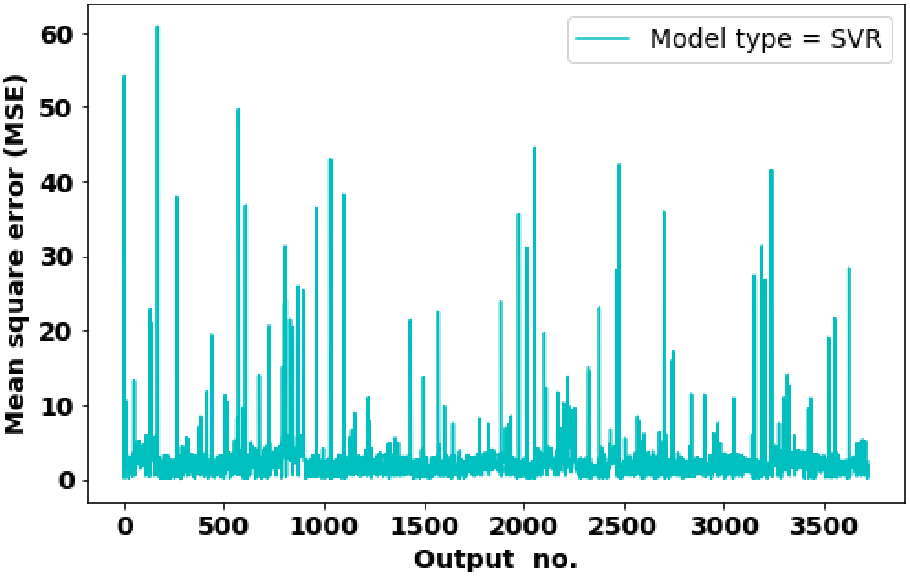


Figure 6. 13 Different output no. versus corresponding mean square error (for test outputs at nodal locations 1 to 3720), estimated through AI/ML (Support-Vector-Machine regression with linear kernel function) model. Each test output time-series covers a portion of the steady-state power operation regime and the entire cool-down regime of a design-basis type loading cycle.

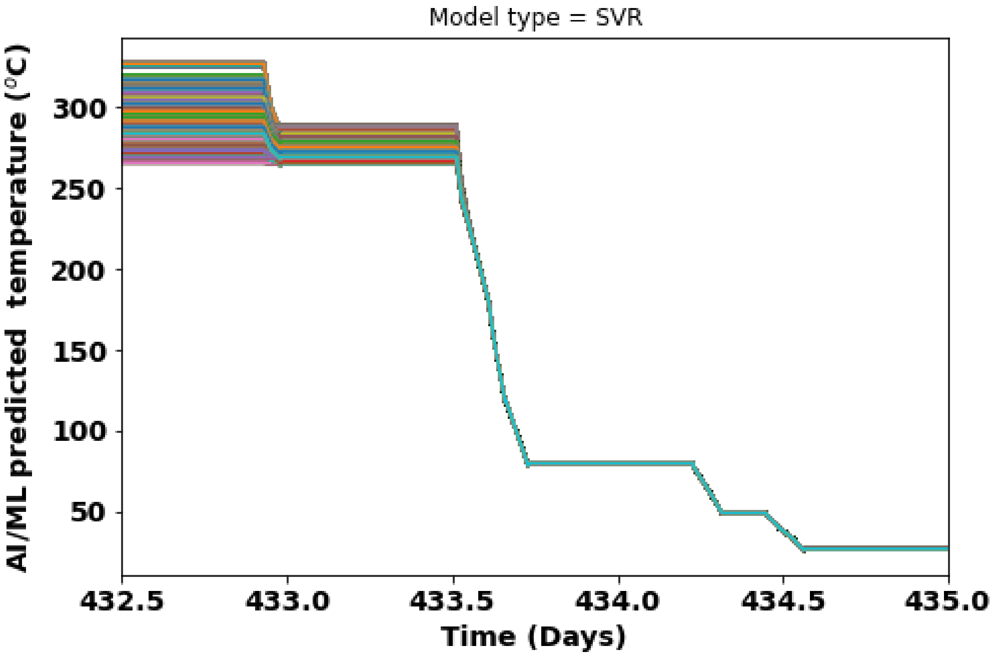


Figure 6. 14 AI/ML (Support-Vector-Machine regression with linear kernel function) model predicted temperatures (or test outputs) at nodal locations 1 to 3720. Each test output time-series covers a portion of the steady-state power operation regime and the entire cool-down regime of a design-basis type loading cycle.

7 Multi-Time-Series Temperature Prediction in a SG-Tube: Predictions at Thousands of 3D Point-clouds Under a Grid-Load-Following (GLF) Loading Cycle

In addition to the model evaluation under DB loading cycle (section 5 and 6), we also performed model evaluation under random grid-load-following (GLF) loading cycle. In this section, we present the related temperature prediction results at thousands of 3D locations in a SG-tube (refer Figure 2.15). The GLF-loading based spatial-temporal temperatures were predicted using the multivariate linear, decision-tree based RF and GB and linear-kernel-function based SVM models. Below presents the details of the training data and model evaluation results.

7.1 Training and Test Data Used for Grid-Load-Following Loading Case

In the discussed prediction cases, the model parameters were not again estimated. Rather the parameters estimated for the predication models discussed in section 6 were used. Note that the models discussed in section 6 were trained using FE simulated data for a DB-loading cycle. The aim of the prediction results presented in section 7 was to check the prediction accuracy under a completely different loading cycle (such as under a GLF loading cycle). If the model predicts accurately under the GLF loading cycle, the associated models can be used for predicting in actual plant with random and real-time process measurements. Figure 7.1 shows the test input/process temperature data (covering an entire grid-load-following type cycle), which were used for evaluating all the discussed AI/ML models in section 7.

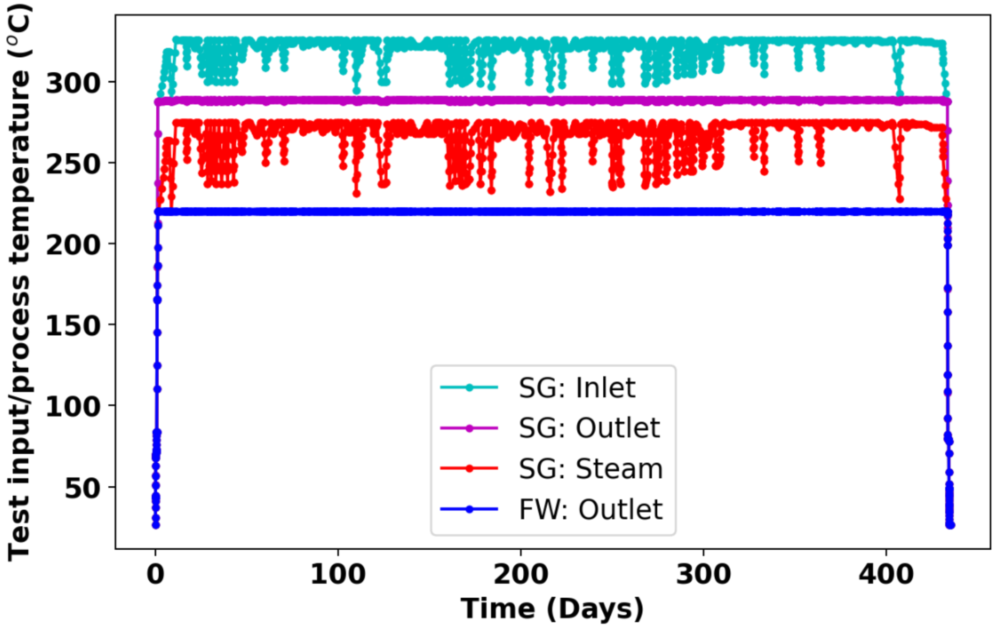


Figure 7. 1 Test-input-temperature data (covering an entire grid-load-following type cycle), which were used for evaluating all the discussed AI/ML models in section 7.

7.2 Multivariate Linear (LR) Regression Model Based Temperature Prediction at Thousands of 3D Point-clouds under Grid-Load-Following (GLF) Loading Cycle

The multivariate linear regression model was used for predicting the temperature of the SG-tube at all its 3720 FE nodes or point-clouds (refer Figure 2.15). The temperatures were predicted under GLF-loading cycle. Figures 7.2 and 7.3 respectively show the training output no. versus training scores R2 and the test output no. versus MSE of test outputs at nodal locations 1 to 3720. Figure 7.4 shows the AI/ML (multivariate linear regression) model predicted temperatures (or test outputs) at nodal locations 1 to 3720. Note that each test/output time-series covers an entire grid-load-following type cycle. Figure 7.5 shows the magnified version of Figure 7.4 showing the AI/ML model predicted temperature during the grid-load-following power operation (i.e., excluding the heat-up and cool-down regime of a reactor fuel cycle). Figures 7.6 and 7.7 show the corresponding actual FE model simulated temperatures. Comparing Figures 7.4 and 7.5 with Figures 7.6 and 7.7 it can be found that the AI/ML based multivariate linear model qualitatively predict good results compared to the FE model. However, looking at Figure 7.3 the MSE for some of the outputs are very high meaning the AI/ML based multivariate linear models are not robust for all combination of process inputs. Hence it is not advisable to use the linear models for this type of DT applications (such as for predicting spatial-temporal temperature in a safety-critical reactor component).

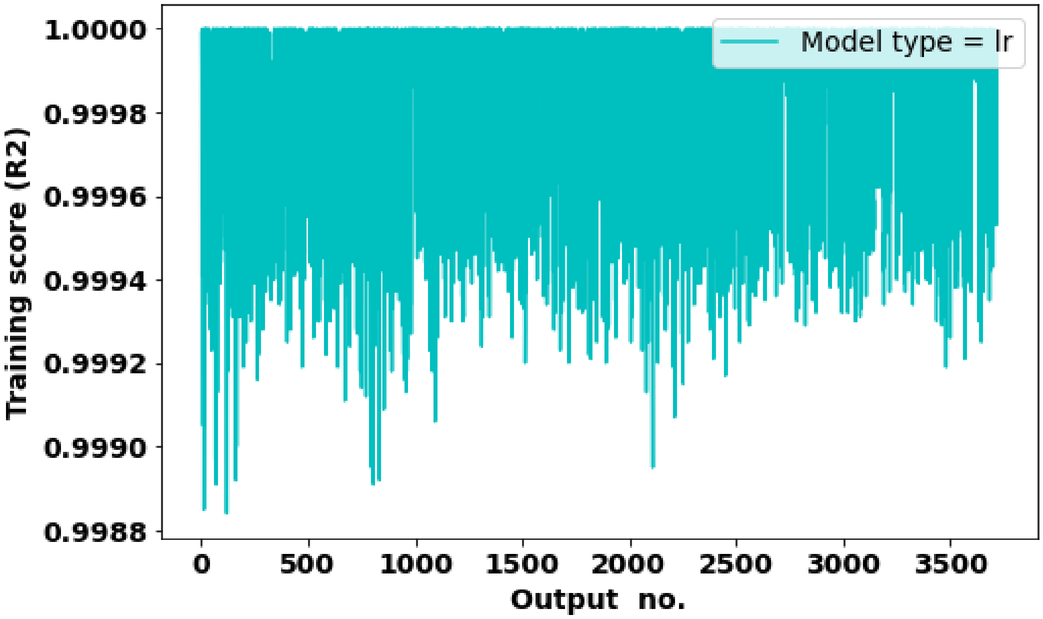


Figure 7. 2 Different output no. versus corresponding training scores R2 (i.e., the individual coefficients of determination for training outputs at nodal locations 1 to 3720), estimated through AI/ML (linear regression) model. Each test/output time-series covers an entire grid-load-following type cycle.

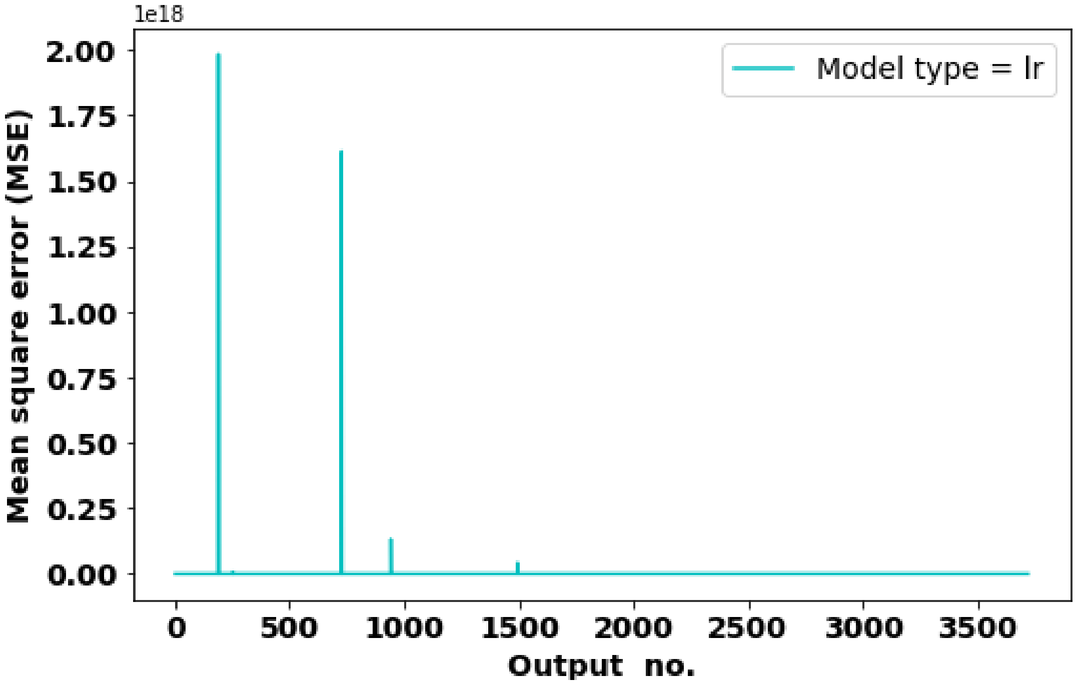


Figure 7. 3 Different output no. versus corresponding mean square error (for test outputs at nodal locations 1 to 3720), estimated through AI/ML (linear regression) model. Each test/output time-series covers an entire grid-load-following type cycle.

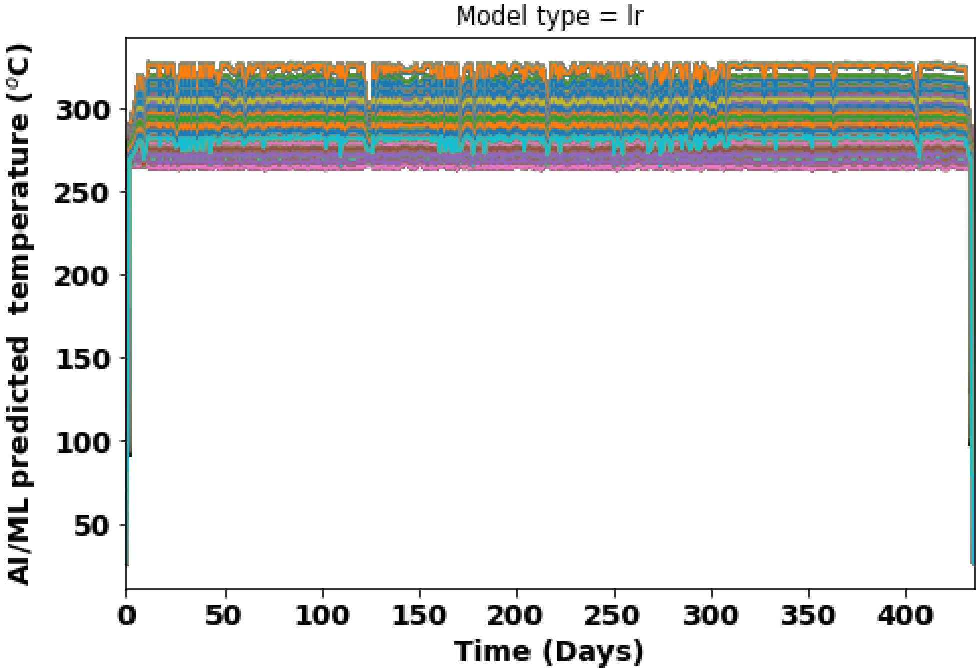


Figure 7. 4 AI/ML (linear regression) model predicted temperatures (or test outputs) at nodal locations 1 to 3720. Each test/output time-series covers an entire grid-load-following type cycle.

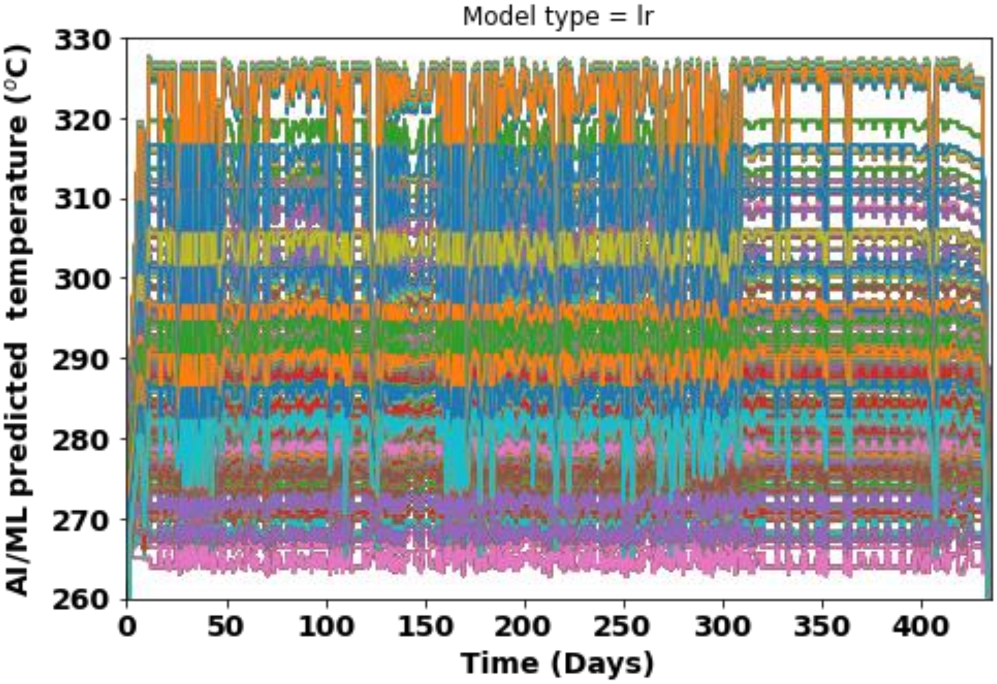


Figure 7. 5 Magnified Figure 7.4 showing the AI/ML model predicted temperature during the grid-load-following power operation.

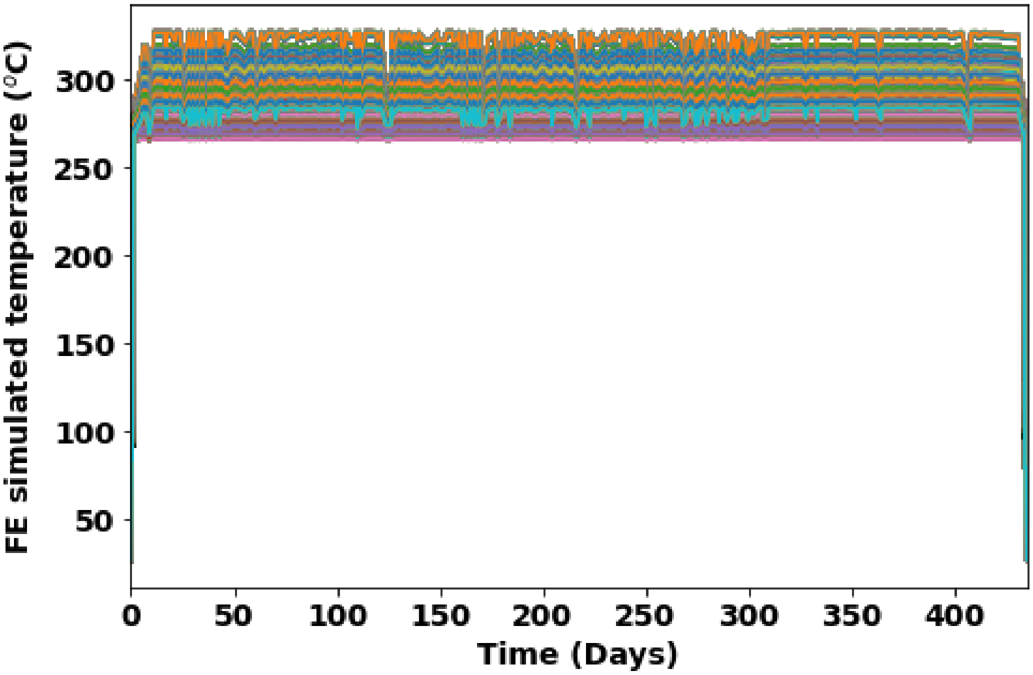


Figure 7. 6 Actual FE model simulated temperatures (or test outputs) at nodal locations 1 to 3720. Each test output time-series covers a portion of the steady-state power operation regime and the entire cool-down regime of a design-basis type loading cycle.

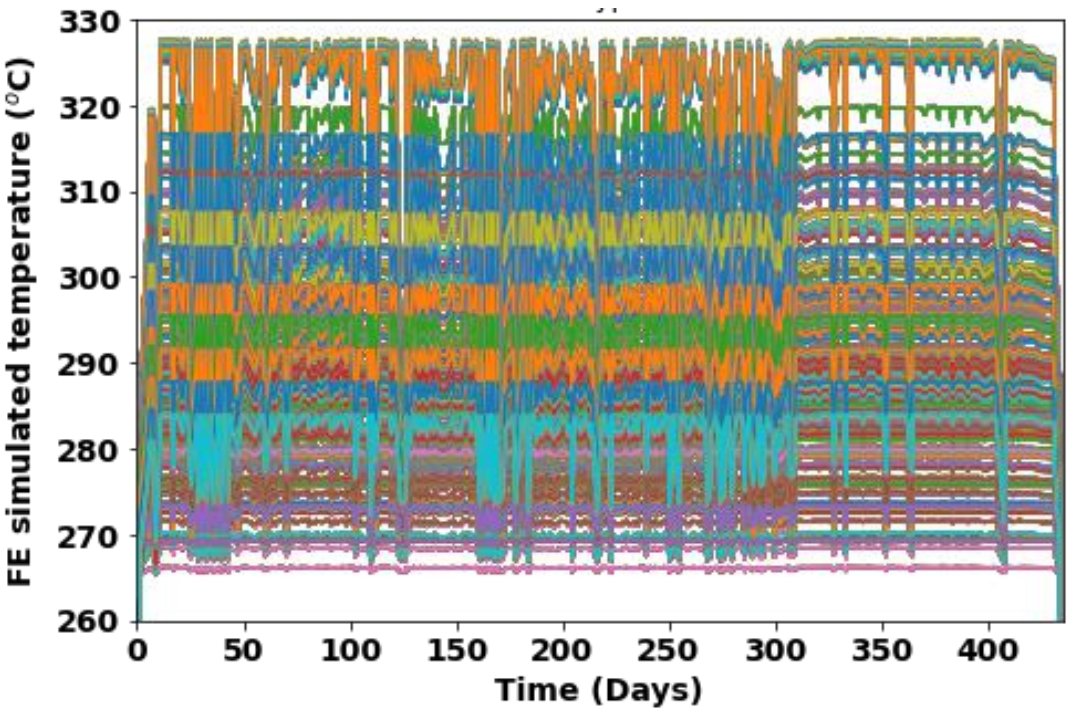


Figure 7. 7 Magnified Figure 7.6 showing the FE model simulated temperature during the grid-load-following power operation.

7.3 Random-Forest (RF) Regression Model Based Temperature Prediction at Thousands of 3D Point-clouds under Grid-Load-Following Loading Cycle

The results related to RF-model based predictions under a GLF loading cycle are presented here. Figures 7.8 and 7.9 respectively show the associated training output no. versus training scores R2 and the test output no. versus MSE of test outputs (at all nodal locations 1 to 3720). Figures 7.10 shows the AI/ML (RF regression) model predicted temperatures at all the spatial locations (node 1 to 3720) of SG-tube shown in Figure 2.15. Figure 7.11 shows the magnified version of Figure 7.10 showing AI/ML model prediction only during the GLF power operation (i.e., excluding the heat-up and cool-down regime). Comparing the RF model predicted spatial-temporal temperatures (Figures 7.10 and 7.11) with corresponding FE model predictions (Figures 7.6 and 7.7) we can see there is a good qualitative matching between RF and FE model predictions. This can be much evident from the MSE plot shown in Figure 7.9 showing a maximum MSE value approximately of 520. This is very similar as the maximum MSE value observed in case of DB loading cycle (refer section 6.3).

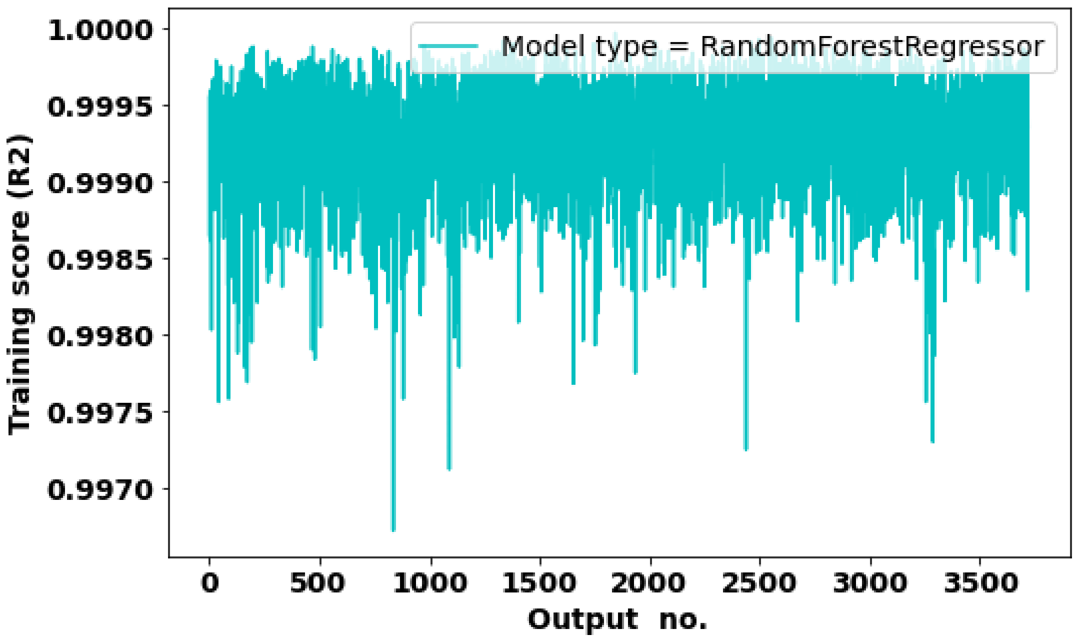


Figure 7. 8 Different output no. versus corresponding training scores R2 (i.e., the individual coefficients of determination for training outputs at nodal locations 1 to 3720), estimated through AI/ML (Random-Forest regression) model. Each test/output time-series covers an entire grid-load-following type cycle.

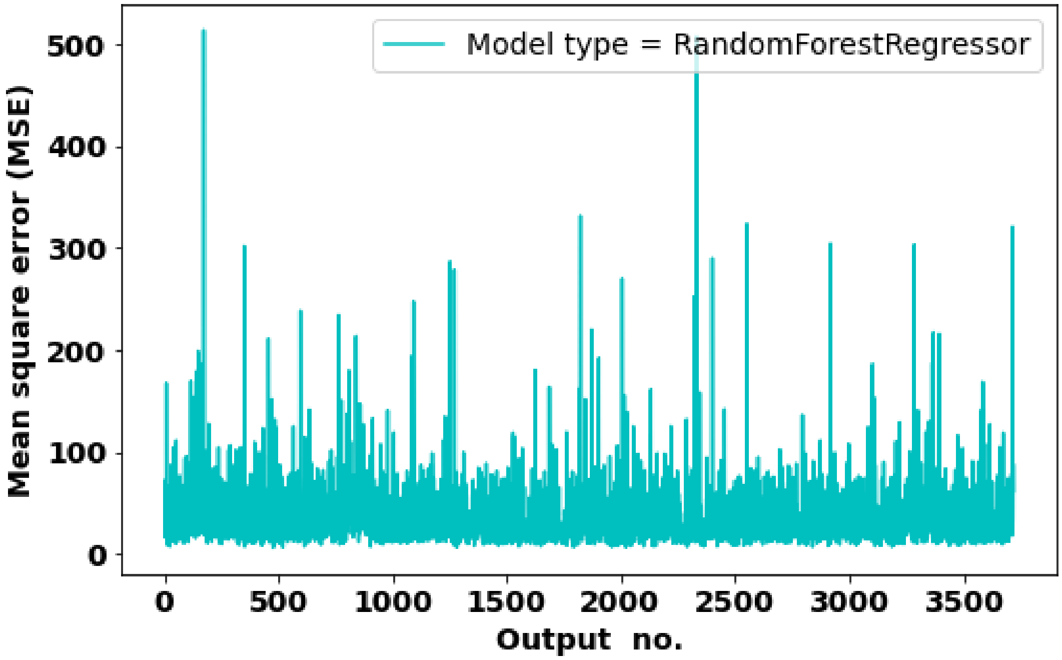


Figure 7. 9 Different output no. versus corresponding mean square error (for test outputs at nodal locations 1 to 3720), estimated through AI/ML (Random-Forest regression) model. Each test/output time-series covers an entire grid-load-following type cycle.

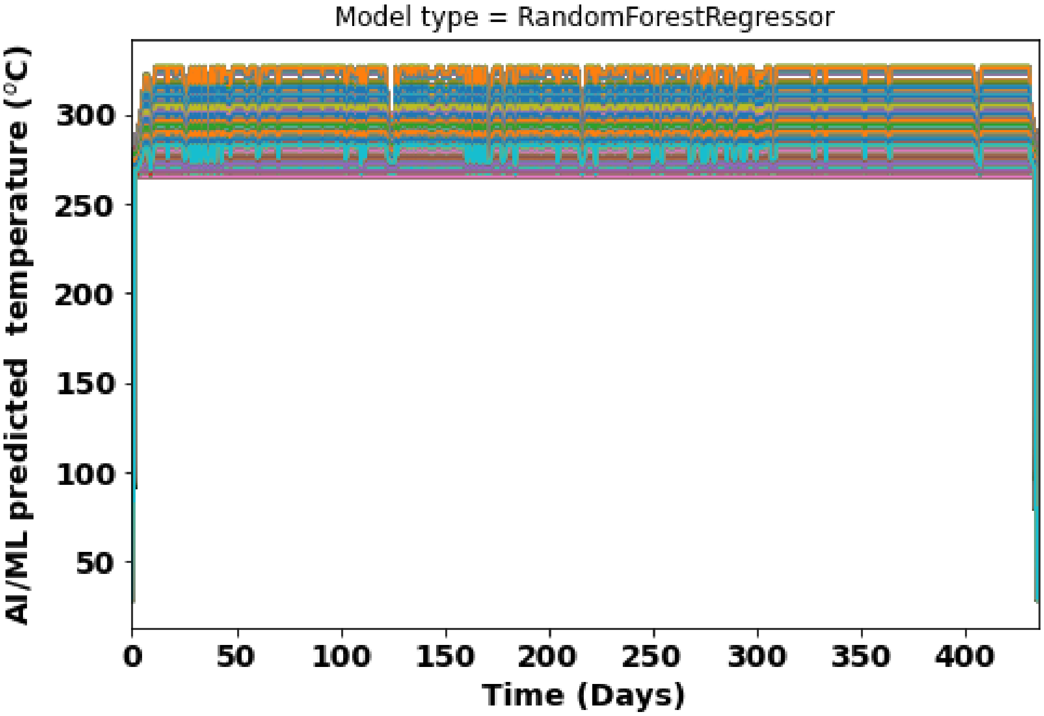


Figure 7. 10 AI/ML (Random-Forest regression) model predicted temperatures (or test outputs) at nodal locations 1 to 3720. Each test/output time-series covers an entire grid-load-following type cycle.

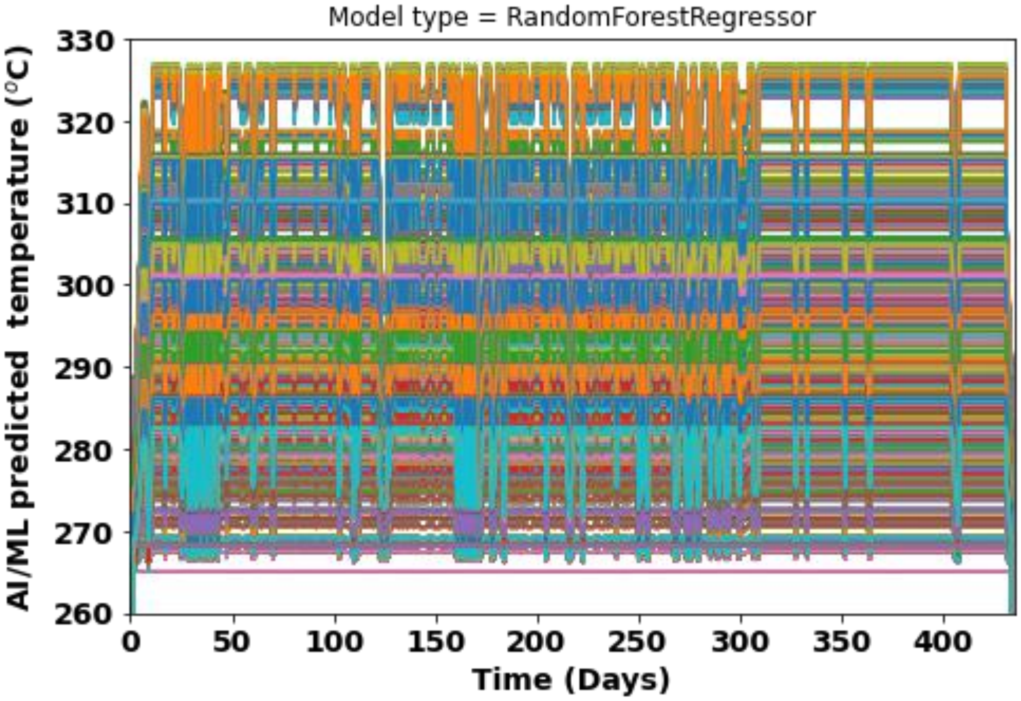


Figure 7. 11 Magnified Figure 7.10 showing the AI/ML model predicted temperature during the grid-load-following power operation.

7.4 Gradient-Boosting (GB) Regression Model Based Temperature Prediction at Thousands of 3D Point-clouds under Grid-Load-Following (GLF) Loading Cycle

The decision-tree based GB-model was evaluated against predicting spatial-temporal temperatures at thousands of locations under a GLF loading cycle. Figures 7.12 and 7.13 show the associated training output no. versus training scores R2 and the test output no. versus MSE of test outputs at all nodal locations 1 to 3720, respectively. Figures 7.14 and 7.15 show the corresponding GB model predicted temperatures (or test outputs) at all the 3D nodal locations highlighted in Figure 2.15. Comparing Figure 7.13 with Figure 6.10, the GB model under GLF loading cycle predicts with better accuracy than under DB loading cycle (i.e., with maximum MSE of 350 versus 520). Nevertheless, the model prediction accuracy for both RF and GB models and both under DB and GLF loading can be considered similar.

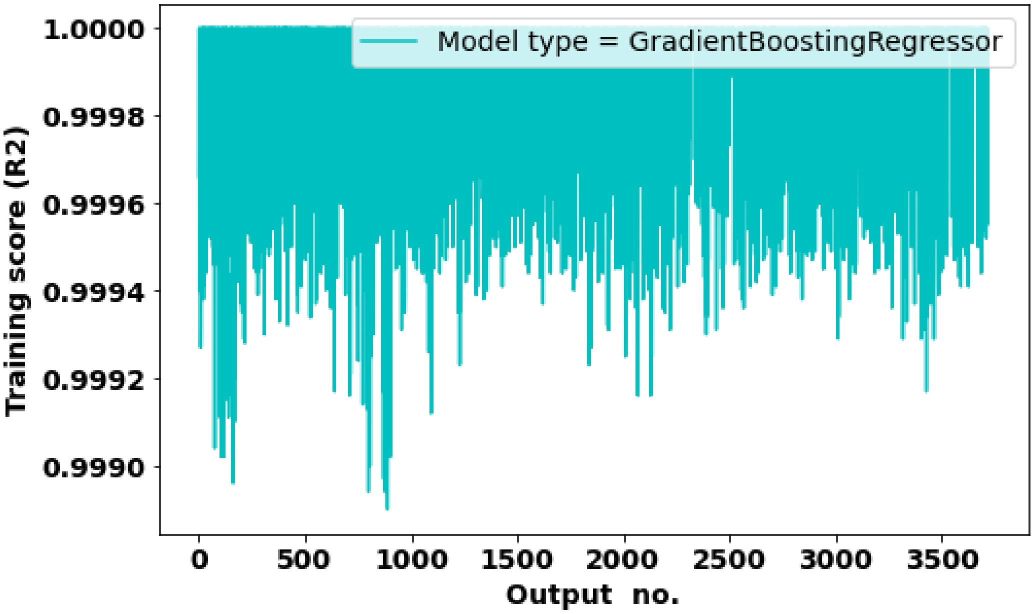


Figure 7. 12 Different output no. versus corresponding training scores R2 (i.e., the individual coefficients of determination for training outputs at nodal locations 1 to 3720), estimated through AI/ML (Gradient-Boosting regression) model. Each test/output time-series covers an entire grid-load-following type cycle.

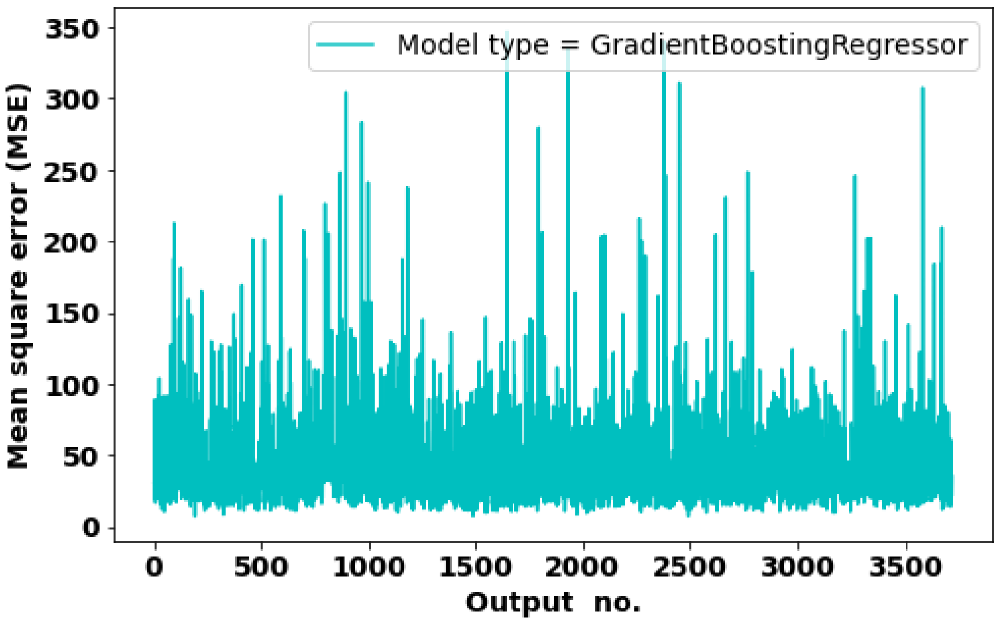


Figure 7. 13 Different output no. versus corresponding mean square error (for test outputs at nodal locations 1 to 3720), estimated through AI/ML (Gradient-Boosting regression) model. Each test/output time-series covers an entire grid-load-following type cycle.

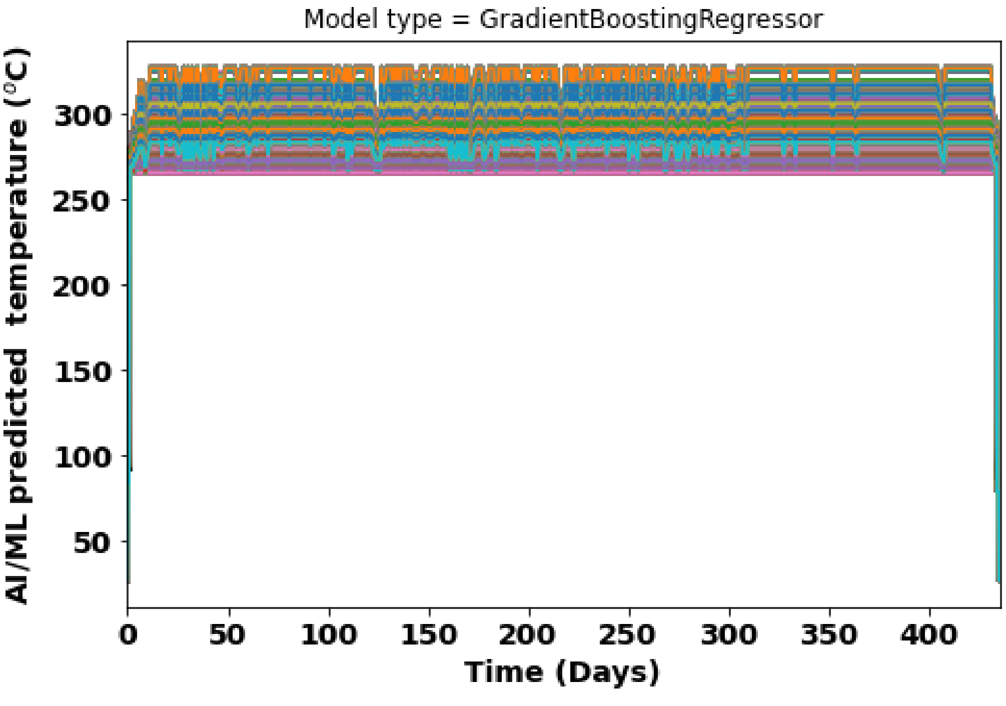


Figure 7. 14 AI/ML (Gradient-Boosting regression) model predicted temperatures (or test outputs) at nodal locations 1 to 3720. Each test/output time-series covers an entire grid-load-following type cycle.

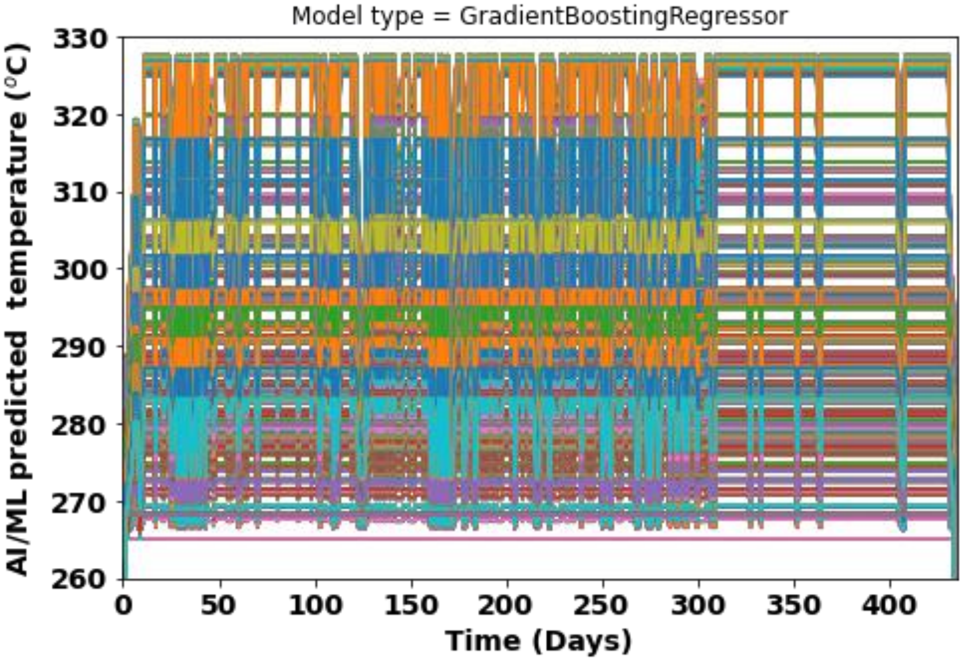


Figure 7. 15 Magnified Figure 7.14 showing the AI/ML model predicted temperature during the grid-load-following power operation.

7.5 Support-Vector-Machine (SVM) Regression Model Based Temperature Prediction at Thousands of 3D Point-clouds under Grid-Load-Following (GLF) Loading Cycle

The SVM-model was evaluated for predicting the spatial-temporal temperatures at thousands of locations under a GLF loading cycle. The SVM model was constructed using the linear kernel function. Figures 7.16 and 7.17 respectively show the associated training output no. versus training scores R2 and the test output no. versus MSE of test outputs at all nodal locations 1 to 3720. Whereas Figures 7.18 and 7.19 show the corresponding AI/ML (SVM regression) model predicted temperatures at all the spatial locations (node 1 to 3720) of SG-tube shown in Figure 2.15. Comparing the SVM model predicted spatial-temporal temperatures (Figures 7.18 and 7.19) with corresponding FE model predictions (Figures 7.6 and 7.7) we can see there is a good qualitative matching between SVM and FE model predictions. In addition, from Figure 7.17 the maximum estimated MSE is of approximately 100, which is very similar number compared to MSE under DB loading cycle (maximum estimated MSE of 65, refer section 6.5). Overall, the decision-tree based RF and GB models and the high-dimensional kernel function based SVM model outperforms the conventional multivariate linear models. Although either of the three models; RF, GB and SVM can be used for the discussed temperature prediction application, the SVM model (with linear kernel function) is the most accurate among all the model evaluated.

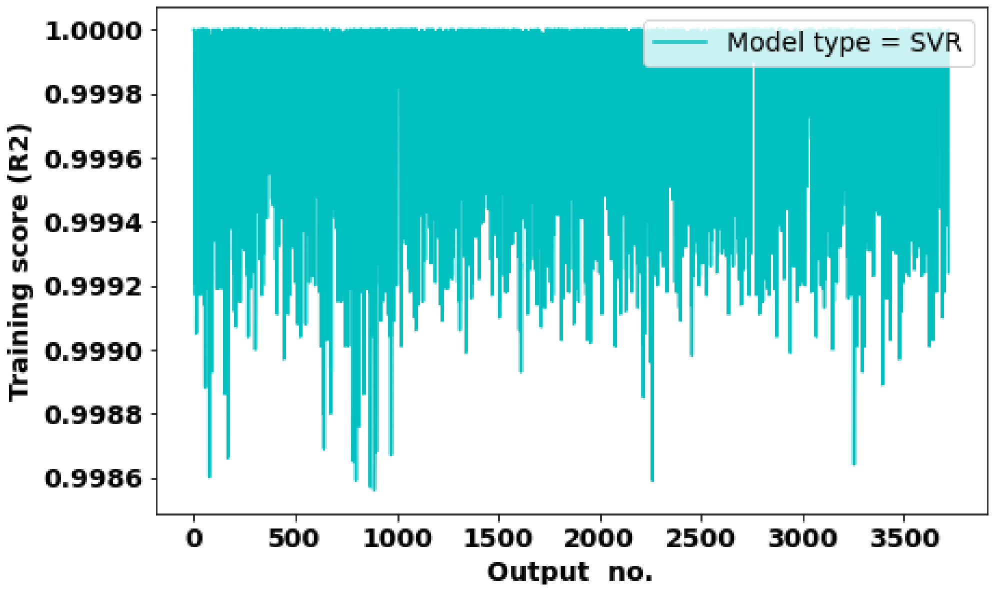


Figure 7. 16 Different output no. versus corresponding training scores R2 (i.e., the individual coefficients of determination for training outputs at nodal locations 1 to 3720), estimated through AI/ML (Support-Vector-Machine regression with linear kernel function) model. Each test/output time-series covers an entire grid-load-following type cycle.

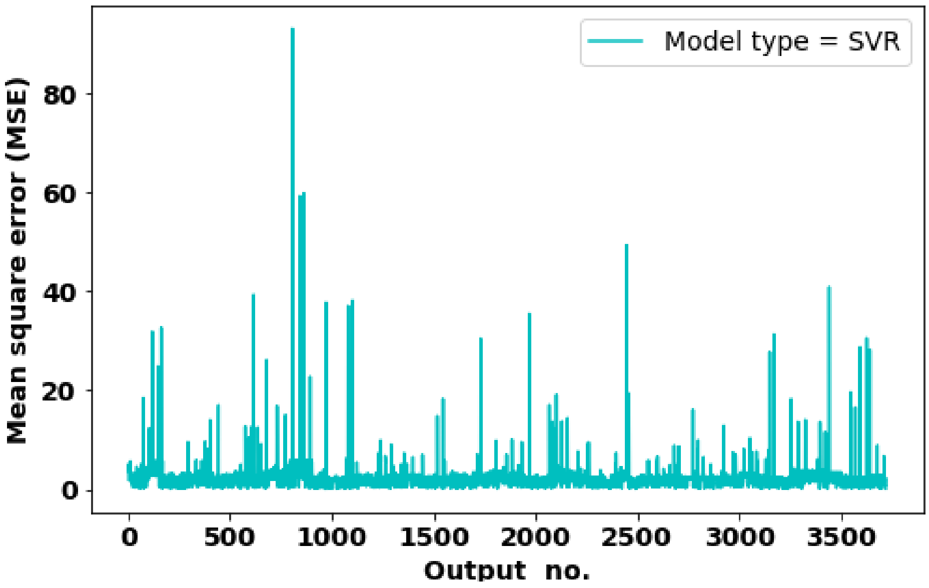


Figure 7. 17 Different output no. versus corresponding mean square error (for test outputs at nodal locations 1 to 3720), estimated through AI/ML (Support-Vector-Machine regression with linear kernel function) model. Each test/output time-series covers an entire grid-load-following type cycle.

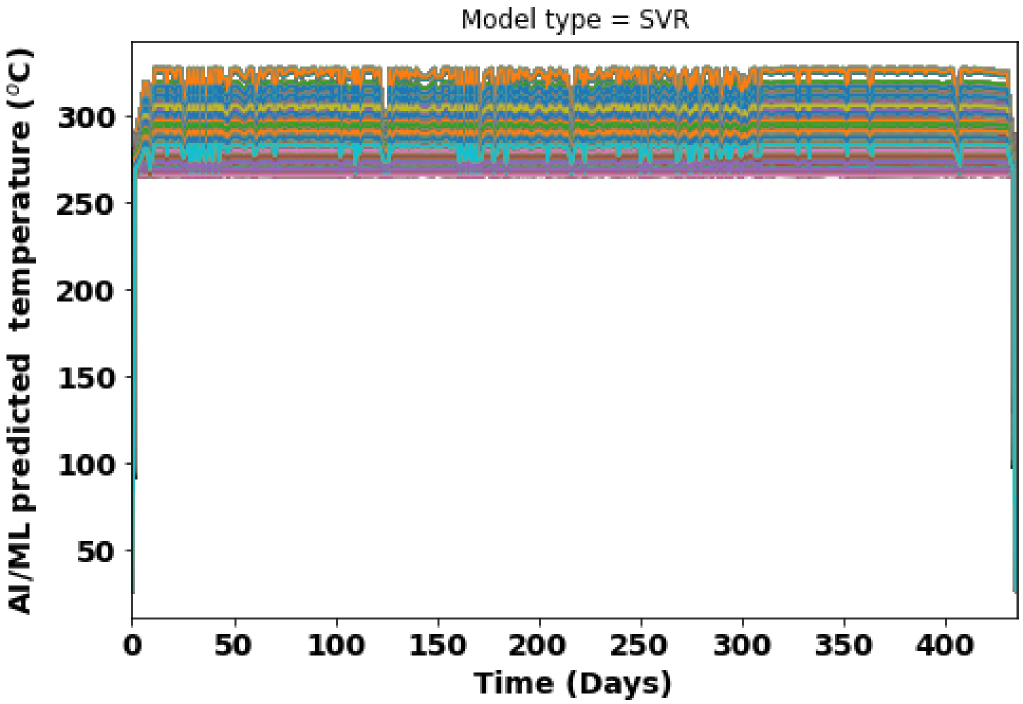


Figure 7. 18 AI/ML (Support-Vector-Machine regression with linear kernel function) model predicted temperatures (or test outputs) at nodal locations 1 to 3720. Each test/output time-series covers an entire grid-load-following type cycle.

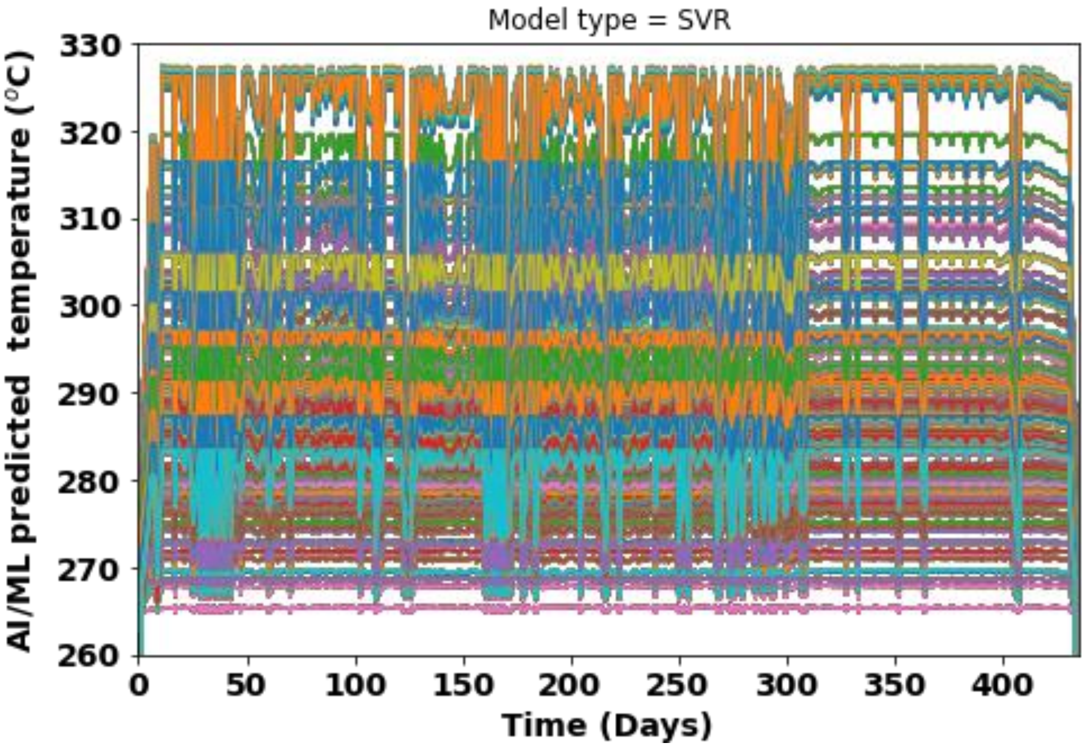


Figure 7. 19 Magnified Figure 7.18 showing the AI/ML model predicted temperature during the grid-load-following power operation.

8 Summary

We are developing an expert-elicitation-digital-twin framework for real time damage state prediction in heat exchanger components of a nuclear reactor. This report presents some of the ongoing work, which are summarized as below:

a) A FE based heat transfer analysis model of an example heat exchanger such as of a PWR steam generator was developed for generating the geometry-dependent virtual sensor data under design-basis and grid-load-following loading cycles. Based on the resulting virtual sensor data various AI/ML models were trained and evaluated.

b) A MongoDB Based NoSQL database implemented to store heterogeneous sensor data (both from virtual and actual sensors).

c) Different AI/ML based multi-time-series predictive models were developed and evaluated against predicting temperature at thousands of 3D point-clouds (in an example component such as in a SG tube). This includes predicting temperature at both inside and outside thickness of a component. The resulting temperature later can be used for predicting the temperature-gradient-dependent stress-strain and associated damage progression in a component.

d) The models were developed based on multivariate linear regression, ensembled decision-tree based Random-Forest (RF) and Gradient-Boosting (GB) regression and high-dimensional-kernel-function-transformation based Support-Vector-Machine (SVM) regression techniques. From the prediction results it has been found that the RF, GB and SVM regression models outperform the multivariate linear regression models. In addition, it is found that the SVM based models (with linear kernel function) are the best among all the models evaluated.

e) The DT models are developed keeping in mind, the model must predict in real-time (e.g., the spatial-temporal temperature, stress, strain, and associated damage states) and with respect to real-time process measurements (from a few existing process measurement sensors).

References

1. Mohanty, S., Park, J.P. and Listwan, J.T., 2019. A System-Level Framework For Fatigue Life Prediction of PWR Pressurizer-Surge-Line Nozzle under Design-Basis Loading Cycles. A complete tensile test-based material properties database and preliminary results on 3D weld process modeling, thermal-mechanical stress analysis and environmental fatigue testing (No. ANL-LWRS-19/01). Argonne National Lab. (ANL), Argonne, IL (United States).
2. Nguyen, T. N., Downar, T., & Vilim, R. (2020). A probabilistic model-based diagnostic framework for nuclear engineering systems. Annals of Nuclear Energy, 149, 107767.
3. Vilim, R., & Nguyen, T. (2021, June). Physics-based Automated Reasoning for Health Monitoring: Sensor Set Selection. In 12th Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies, virtual meeting.
4. Nguyen, T. N., Vilim, R., and Ponciroli, R., “Fault Diagnosis Framework for Standalone Component,” patent application, filed U.S. Patent Office, December 2020.
5. ASME Boiler and Pressure Vessel Code, II – Materials, 2017 version, The American Society of Mechanical Engineers.
6. Lokhov, A. "Load-following with nuclear power plants." NEA News 29, no. 2 (2011): 18 (https://www.oecd-nea.org/nea-news/2011/29-2/nea-news-29-2-load-following-e.pdf).
7. Codd, E. F. (1970). "A Relational Model of Data for Large Shared Data Banks". Communications of the ACM. 13 (6): 377–387. doi:10.1145/362384.362685
8. Gray, Jim. "The transaction concept: Virtues and limitations." In VLDB, vol. 81, pp. 144-154. 1981.
9. Haerder, Theo, and Andreas Reuter. "Principles of transaction-oriented database recovery." ACM computing surveys (CSUR) 15, no. 4 (1983): 287-317.
10. https://www.oracle.com/database/
11. https://www.mysql.com/
12. https://www.microsoft.com/en-us/sql-server?rtc=1
13. Moniruzzaman, A. B. M., and Syed Akhter Hossain. "Nosql database: New era of databases for big data analytics-classification, characteristics and comparison." arXiv preprint arXiv:1307.0191 (2013).
14. Abramova, Veronika, Jorge Bernardino, and Pedro Furtado. "Which nosql database? a performance overview." Open Journal of Databases (OJDB) 1, no. 2 (2014): 17-24.
15. Decandia, G.; Hastorun, D.; Jampani, M.; Kakulapati, G.; Lakshman, A.; Pilchin, A.; Sivasubramanian, S.; Vosshall, P.; Vogels, W. (2007). "Dynamo: Amazon's Highly Available Key-value Store". Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles - SOSP '07. p. 205. doi:10.1145/1294261.1294281
16. Weintraub, Grisha. "Dynamo and BigTable—Review and comparison." In 2014 IEEE 28th Convention of Electrical & Electronics Engineers in Israel (IEEEI), pp. 1-5. IEEE, 2014.
17. Chang, Fay, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber. "Bigtable: A distributed storage system for structured data." ACM Transactions on Computer Systems (TOCS) 26, no. 2 (2008): 1-26.

18. Huang, Hongcheng, and Ziyu Dong. "Research on architecture and query performance based on distributed graph database Neo4j." In 2013 3rd International Conference on Consumer Electronics, Communications and Networks, pp. 533-536. IEEE, 2013.
19. https://neo4j.com/
20. Wei-Ping, Zhu, L. I. Ming-Xin, and Chen Huan. "Using MongoDB to implement textbook management system instead of MySQL." In 2011 IEEE 3rd International Conference on Communication Software and Networks, pp. 303-305. IEEE, 2011.
21. https://www.mongodb.com/
22. Han, Jing, Ee Haihong, Guan Le, and Jian Du. "Survey on NoSQL database." In 2011 6th international conference on pervasive computing and applications, pp. 363-366. IEEE, 2011.
23. https://couchdb.apache.org/
24. https://www.dnsstuff.com/nosql-database-comparison
25. https://www.improgrammer.net/most-popular-nosql-database/
26. Mohanty, S., Jagielo, B., Iverson, W., Bahn, C., Soppet, W. K., Majumdar, S., and Natesan, K., 2014, "Online Stress Corrosion Crack and Fatigue Usages Factor Monitoring and Prognostics in Light Water Reactor Components: Probabilistic Modeling, System Identification and Data Fusion Based Big Data Analytics Approach," Argonne National Laboratory Report ANL/LWRS-14/02, https://www.osti.gov/biblio/1168230 [doi: 10.2172/1168230] .
27. Mohanty, S., 2010. Structural health monitoring and condition based fatigue damage prognosis of complex metallic structures. Arizona State University.
28. Mohanty, S., Park, J.P. and Listwan, J.T., 2019. A System-Level Framework for Fatigue Life Prediction of PWR Pressurizer-Surge-Line Nozzle under Design-Basis Loading Cycles. A complete tensile test based material properties database and preliminary results on 3D weld process modeling, thermal-mechanical stress analysis and environmental fatigue testing Argonne National Laboratory Report ANL-LWRS-19/01, https://www.osti.gov/biblio/1571258 [doi: 10.2172/1571258].
29. Szilard, R., Planchon, P. and Busby, J., 2009. The case for extended nuclear reactor operation. JOM, 61(7), pp.24-27.
30. https://pymongo.readthedocs.io/en/stable/
31. Zhang, X., Song, W. and Liu, L., 2014, June. An implementation approach to store GIS spatial data on NoSQL database. In 2014 22nd international conference on geoinformatics (pp. 1-5). IEEE.
32. Scikit-learn documentation, https://scikit-learn.org/stable/
33. Quinlan, J. Ross. C4. 5: programs for machine learning. Elsevier, 2014.
34. Breiman, Leo, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. Classification and regression trees. Routledge, 2017.
35. Breiman, "Random Forests", Machine Learning, 45(1), 5-32, 2001.
36. Friedman, Jerome H. "Greedy function approximation: a gradient boosting machine." Annals of statistics (2001): 1189-1232.
37. Han, Jiawei, Jian Pei, and Micheline Kamber. Data mining: concepts and techniques. Elsevier, 2011.
38. Smola, Alex J., and Bernhard Schölkopf. "A tutorial on support vector regression." Statistics and computing 14, no. 3 (2004): 199-222.

Nuclear Science and Engineering Division
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 208
Argonne, IL 60439

www.anl.gov`

Argonne National Laboratory is a U.S. Department of Energy
laboratory managed by UChicago Argonne, LLC