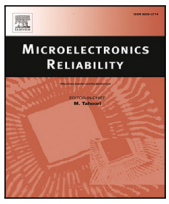
Microelectronics Reliability 174 (2025) 115900

Contents lists available at ScienceDirect

Microelectronics Reliability

journal homepage: www.elsevier.com/locate/microrel



Research paper

Application of machine learning modeling for predicting the reliability of solder joints under thermal cycling

Qiulin Yu [a,b,e,*], Chinmay Nawghane [c], Zihan Zhang [d], Bart Vandevelde [c], Karl Fendt [a], Thomas Krivec [a], Dieter P. Gruber [b,e]

[a] Austria Technologie & Systemtechnik Aktiengesellschaft, Fabriksgasse 13, Leoben, 8700, Styria, Austria
[b] Chair of Materials Science and Testing of Polymers of Montanuniversitaet, Otto-Glöckel-Straße 2, Leoben, 8700, Styria, Austria
[c] Imec, Kapeldreef 75, Leuven, 3001, Flemish Brabant, Belgium
[d] Department of Microelectronics, Delft University of Technology, Mekelweg 5, Delft, 2628 CD, Netherlands
[e] Polymer Competence Center Leoben GmbH, Sauraugasse 1, Leoben, 8700, Styria, Austria

ARTICLE INFO

Keywords:
Machine learning
Microelectronics
Solder joint
Thermal cycling
Surrogate modeling
Regression analysis

ABSTRACT

In this study, Machine Learning (ML) methods combined with Optuna hyperparameter optimization were investigated to predict creep strain in solder joints of multilayer chip capacitors. Material properties, geometry and thermal loading conditions were varied in simulations using Finite Element Modeling. Evaluated ML models included Random Forest, Gradient Boosting, Support Vector Regression (SVR) and Artificial Neural Network (ANN). The results demonstrated a prediction accuracy of 96%, particularly for SVR and ANN. The model performance significantly improved with increasing data size up to around 600 simulations. In the feature and hyperparameter importance analysis, solder stand-off height and component length most influenced ANN predictions, with learning rate being the key hyperparameter, while for SVR, the regularization parameter or kernel function was most critical.

1. Introduction

Solder joints in electronic packaging endure substantial mechanical and thermal stresses during operation. Due to the low strength and elastic modulus of solder materials, solder joints are typically the most vulnerable components and are highly susceptible to low cycle fatigue (LCF) [1,2]. The fatigue failure of a solder joint can generally be divided into two main stages: crack initiation and crack propagation. Crack initiation generally follows three stages: the formation of a microcrack, its nucleation, and finally, the initiation of a significant physical crack [3–5]. In some cases, cracks can form due to overloading, where an accidental force or pre-existing void in the crack region triggers their development. However, in fatigue testing, crack initiation primarily results from the cyclic deformation of the solder joint under stress [6]. During exposure to thermal cycling, thermomechanical stresses and strains are generated due to the mismatch in the coefficient of thermal expansion (CTE) between the printed circuit board (PCB) and electronic components, which drive progressive creep deformation and eventually result in fatigue failure of the solder joints [7,8]. Over the product's life cycle, these repeated thermal cycles contribute to crack initiation and propagation, gradually degrading the mechanical integrity of the solder joints and reducing the overall reliability of the electronic assembly.

To assess and predict the long-term durability of solder joints under real-world operating conditions, accelerated thermal cycling tests are commonly employed [9]. These tests subject electronic components to rapid and extreme temperature fluctuations, replicating years of thermal cycling within a significantly shorter period. By analyzing the failure mechanisms and fatigue life observed in accelerated thermal cycling tests, engineers can develop predictive models to estimate the actual service life of solder joints and optimize material and design choices to enhance reliability [10]. However, experimental testing for electronic package typically takes 3 to 4 months to complete, requiring significant costs and resources. This is where Finite Element Modeling (FEM) comes into play, providing a more efficient way to model complex engineering problems. A proper Finite Element (FE) model with large nonlinearities of material and geometry usually requires specialized modeling expertise of researchers and high computational power of hardware device [11–13].

In order to overcome this issue, the integration of artificial intelligence (AI) techniques with FEM simulations has emerged as an effective approach for achieving reliable predictions in the microelectronics industry [14–17]. This methodology, referred to Fig. 1, is known as the AI-assisted design-on-simulation procedure. It involves varying

* Corresponding author at: Austria Technologie & Systemtechnik Aktiengesellschaft, Fabriksgasse 13, Leoben, 8700, Styria, Austria.
E-mail address: qiulin.yu@pccl.at (Q. Yu).

https://doi.org/10.1016/j.microrel.2025.115900
Received 16 May 2025; Received in revised form 6 August 2025; Accepted 27 August 2025
Available online 6 September 2025

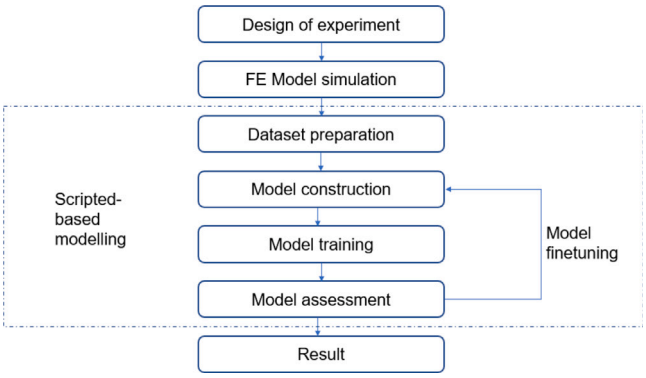


Fig. 1. Architecture of AI-assisted design-on-simulation procedure.

geometry, material parameters and boundary conditions throughout the simulations to cover the design space and represent real scenarios. A comprehensive dataset is then generated via FEM simulations for AI training purposes. Once the model is developed, designers only need to input the required parameters to obtain the target values [18]. Several studies have employed this methodology for assessing solder joint reliability, which will be further discussed in Section 2. However, most existing studies use only one type of modeling approach, and there is a lack of comparative analysis between different machine learning models for predicting solder joint reliability. Additionally, the selection and optimization of hyperparameters present a significant challenge, as improper tuning can lead to suboptimal model performance [19]. Another critical limitation is the uncertainty regarding the optimal dataset size for FEM-generated data. Due to this uncertainty, researchers often generate large datasets to ensure sufficient training samples. However, the marginal benefit of increasing dataset size beyond a certain threshold remains unclear, and excessive data collection may result in unnecessary computational costs without a proportional improvement in model accuracy. Despite these challenges, there is a lack of systematic investigations into how different machine learning models respond to varying dataset sizes and whether their predictive performance stabilizes beyond a certain data volume.

Therefore, this study aims to conduct a comprehensive comparison of various machine learning algorithms, utilizing Optuna as an efficient hyperparameter optimization framework. Furthermore, it systematically examines the relationship between dataset size and model performance, assessing whether increasing the volume of FEM-generated data consistently enhances predictive accuracy or if performance gains diminish beyond a critical threshold. The findings of this study will contribute to a deeper understanding of the trade-off between data volume and predictive efficiency, providing valuable insights for optimizing FEM-based AI modeling in the reliability assessment of solder joints.

2. Predictive modeling techniques in solder joints

Several studies have investigated methods to predict life cycles of solder joint in early design stage and understand how package design parameters affect the solder joint integrity. The most commonly used solder joint fatigue models are creep strain-based and energy-based life models, which are developed by integrating finite element modeling with empirical reliability test data [20–23]. Schubert et al. demonstrated the impact of various solder interconnect alloys and package types on the fatigue life of solder through both simulations and experimental studies [22]. Liu et al. presented a hybrid method that combines analytical and energy-based approaches to predict solder joint geometry and analyze thermal stress/strain behavior in flip-chip packaging [24]. The results demonstrate that optimizing solder ball size and pad configuration can significantly reduce thermal stress, enhancing the reliability of solder joints in advanced electronic packaging applications. Liu et al. further applied this combined method to predict standoff heights and geometry profiles of the solder joints, which is further validated by experiments [25]. Xu et al. utilized finite element simulations and response surface approximation to evaluate the thermo-mechanical performance in the design optimization procedure [26]. Their findings consistently demonstrated a strong correlation between simulation outcomes and experimental data . This agreement reinforces the validity of simulation-based methodologies and provides a robust ground truth for training machine learning models in predictive reliability assessment.

In recent years, ML-based modeling techniques have been widely used in various research domains of electronics [27–29]. In particular, the combination of machine learning and simulation has significantly reduced both cost and time, demonstrating strong ability to analyze complex systems [15]. Yuan et al. combined both the recurrent neural network (RNN) and the gate-network long short-term memory (LSTM) with 81 FE data pairs to assess the solder joint thermal cycling performance in Glass Wafer Level Chip Scale Package, where die thickness, glass thickness and PI thickness are considered [30]. Ferrando-Villalba et al. developed an artificial neural network to predict inelastic strain in PCB Assembly solder joint under thermal cycling [31]. A total of 1017 parameterized simulations were conducted, incorporating 15 variables across various scenarios, including substrate/solder/die geometries, material properties of the substrate and solder, and pitch variations. The ANN predictions exhibited a standard deviation of approximately 15% compared to the simulated strain values.

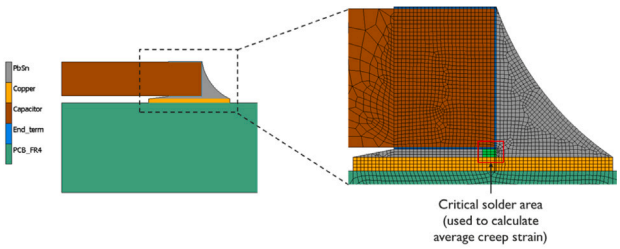
Although machine learning based techniques have been widely applied to predict solder joint reliability, researchers typically rely on extensive simulations to generate training data and spend significant time manually searching for optimal hyperparameters. This work aims to compare various models and develop a framework that simplifies hyperparameter tuning while also investigating the relationship between dataset size and model performance, ultimately reducing the resources required for simulation and modeling in the hyperparameter tuning.

3. Methodology

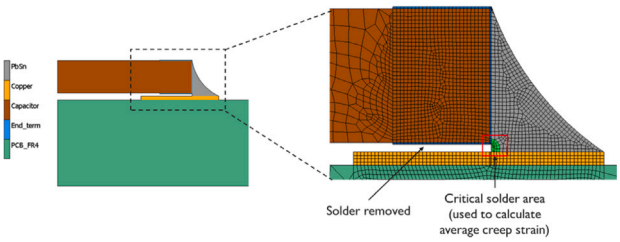
3.1. Dataset generation via finite element simulations

The primary goal of this research is to develop a machine learning (ML) model capable of predicting the creep strain in solder joints of multilayer chip capacitors (MLCC). To achieve this, a substantial amount of data is imperative, which is subsequently generated using FE simulations. We use finite-element modeling to simulate a standard MLCC electronics package with solder joints. The input parameters for these simulations include various material properties, geometric configurations, loading conditions, and environmental variables, all of which are chosen based on their significance to the solder joint performance. These parameters are systematically varied to create a comprehensive dataset. The key output variable from these simulations is the average creep strain measured under different scenarios, which serves as the target variable for our ML model.

The FEM structure was generated with a bottom-up approach using the commercial software MSC Marc [32]. This method involves building the model from the most basic elements such as points, lines, 2D planes and expanding them to a 3D structure. A parametric script is developed to build such FE models, which allows good control over the mesh and enables the definition of consistent mesh even with different geometry parameters. This ensures that the mesh quality remains high and uniform, regardless of the variations in the geometric configurations of the MLCC package. The parametric scripting approach enhances flexibility and efficiency in modeling, as it automates the generation of the mesh based on predefined parameters. This automation reduces the likelihood of human error and ensures consistency across multiple simulations. By adjusting the geometric parameters within the



(a) DASC model: Solder is present both under the component and within the fillet. Crack propagation occurs under the component area, specifically targeting the solder joint located beneath the chip component.



(b) FOSC model: Solder is present only in the solder fillet area. It assumes that the solder joint underneath the component has completely cracked, so that solder below the component is removed.

Fig. 2. Illustration of DASC model and FOSC model with showing mesh and critical solder area.

script, various configurations can be quickly modeled and analyzed, thereby accelerating the dataset generation process. Furthermore, the parametric script facilitates the exploration of a wide range of scenarios, as it can systematically vary input parameters to study their effects on the performance of the solder joints. This comprehensive exploration is crucial for creating a robust dataset that accurately captures the behavior of the components under diverse conditions. Ultimately, this methodological approach significantly contributes to the development of a reliable and predictive machine learning model by providing high-quality and consistent simulation data.

To reduce computational time and generate a large dataset efficiently, the initial 3D model is simplified to a 2D model. This simplification is achieved by applying structural plane strain geometry properties to the 2D model, which effectively provides the necessary thickness to the model without the need for a full 3D representation. This approach is particularly advantageous because it simplifies the complex 3D problems into more manageable 2D problems, allowing for faster computations and easier handling of the data. By simplifying the 3D model to a 2D model, the initial computational time was significantly reduced from 4 h to just 15 min in our simulation attempts. This drastic reduction in computation time enabled the efficient generation of a large dataset, which is essential for training robust machine learning models.

It is also important to note that FE simulations are performed in two essential steps to obtain accurate results. The first step involves crack propagation under the component area, specifically targeting the solder joint located below the chip component. The second step addresses crack propagation within the solder fillet area. To effectively implement this process, two distinct FE models are developed. In Dual-area Solder Coverage (DASC) model, solder is present both under the component and within the fillet. In contrast, Fillet-Only Solder Coverage (FOSC) model assumes a fully cracked solder joint below the component, thus removing the solder from beneath the component area. Fig. 2(a) illustrates DASC model, while Fig. 2(b) depicts FOSC model, along with their respective meshes. The mesh in these models is carefully refined in the solder joint area to accurately capture the creep strain. To maintain consistency across a large number of simulations, a critical solder area is defined. The average creep strain is calculated from this precisely defined area, ensuring uniformity and reliability in the data. This calculated average creep strain is subsequently used as the target variable in the machine learning model, which aims to predict the performance and behavior of solder joints under various conditions.

3.2. Finite element modeling of MLCC

3.2.1. Input parameter definition

The input features selected for this study encompass a variety of material properties, geometric configurations, and loading conditions. The selection of these features is motivated by their critical influence on the performance and reliability of solder joints in MLCCs. The corresponding parameter ranges were determined based on manufacturer datasheets, relevant industry standards, and expert knowledge drawn from prior studies and practical experience.

(a) *Material Properties*
Material properties play a crucial role in determining the mechanical behavior and durability of solder joints under different operating conditions. Key material properties considered in this study include the coefficient of thermal expansion, which reflects the mismatch between component and PCB. This mismatch can induce thermal stresses, leading to creep strain and eventual failure of the solder joints. Elastic/Young's modulus (E-modulus), representing the stiffness of the materials involved, affects the stress distribution and deformation behavior of the solder joints.

(b) *Geometric Configurations*
Geometric parameters are crucial in defining the physical dimensions and structural characteristics of the MLCC package and solder joints. This study includes standard MLCC sizes ranging from C1005 to C7563, based on manufacturer datasheets, to ensure model generalizability across varying capacitor dimensions. Variations in solder stand height impact the overall height of the solder joint, influencing its thermal and mechanical behavior. The length of the copper pads affects the fillet formation of the solder joints, which in turn influences the stress distribution and deformation behavior.

(c) *Loading Conditions*
Loading conditions simulate the operational environment and thermo-mechanical stresses experienced by the solder joints. The minimum and maximum temperatures are defined as critical input parameters, encompassing the standard thermal cycling conditions ranging from −55 °C to 100 °C. This range encompasses representative temperature windows (e.g., −40 °C to 100 °C, 0 °C to 100 °C) aligning with industry standards like ECSS-Q-ST-70-04C for space applications and AEC-Q007-001 for automotive contexts. As such, the model ensures broad applicability across various real-world scenarios.

Table 1 provides a detailed list of the parameters along with the range of values simulated in this study. The geometric parameters, crucial for accurately depicting the physical dimensions and structural characteristics of the MLCC package, are visually represented in Fig. 3. The design of experiments for this study was conducted using the Latin hypercube sampling method. This method is effective in this context as it ensures a comprehensive and uniform exploration of the design space, thereby reducing the number of simulations needed to achieve accurate results. As shown in Fig. A.13, the distribution of all input parameters are illustrated by their histograms. A total of 1481 simulations were conducted for each of the two models, resulting in an aggregate of 2962 simulations. The simulation process was automated using Noesis Optimus software [33], which allows setting up the design of experiments based on the latin hypercube method and performs the simulations sequentially. The study focused on 13

Table 1
List of input parameters used in the simulations along with their ranges.

Type	Input parameters	Nominal	Min	Max
Geometry	Component length (lc)	4.5 mm	0.6	7.5
	Component width (wc)	3.2 mm	0.3	6.3
	Component thickness (tc)	2.8 mm	0.3	3
	End termination length (let)	0.61 mm	0.1	0.7
	Solder stand-off height (tsf)	0.045 mm	0.01	0.07
	Partial Copper pad length (lsf1)	0.3 mm	0.1	0.4
	Partial Copper pad length (lsf2)	0.35 mm	0.2	0.6
Material property	Component CTE (Cap_CTE)	9 ppm/°C	5	10
	PCB CTE X/Y (PCB_CTE)	15 ppm/°C	12	18
	PCB thickness (tpcb)	1.6mm	0.8	3.2
	PCB young's modulus (PCB_E)	23 000 MPa	23 000	50 000
Loading profile	Min temperature (Tmin)	−55 °C	−55	20
	Max temperature (Tmax)	100 °C	80	120

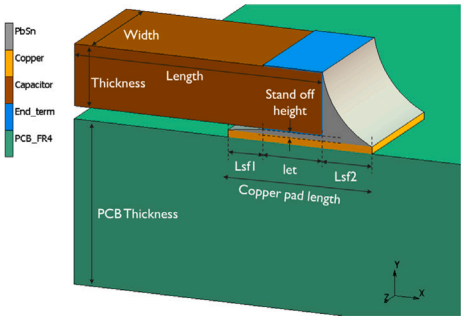


Fig. 3. Schematics of MLCC with geometrical parameters.

input variables, each significantly influencing the outcome. These input variables produced two main outputs: creep strain in DASC model and creep strain in FOSC model. By carefully selecting these input features, the study aims to create a comprehensive and representative dataset that captures the key factors influencing the creep strain in solder joints. This approach ensures that the machine learning model developed can accurately predict the performance of solder joints under diverse conditions, thereby enhancing the reliability and robustness of the MLCC electronics package.

3.2.2. Output parameter definition

The study identifies two primary output parameters: average creep strain from DASC model and average creep strain from FOSC model. In Fig. 4, the results of the finite element (FE) simulation are depicted, showcasing the creep strain observed in each FE model for a full thermal cycle of −55 °C to 100 °C. The Average creep strain is calculated from the critical areas defined for each model, as previously illustrated in Fig. 2. The average creep strain obtained from these simulations serves as the target parameter in the subsequent ML models.

3.3. Dataset visualization and preprocessing

3.3.1. Correlation matrix

A correlation matrix is a tabular representation that displays correlation coefficients, indicating the strength and direction of linear

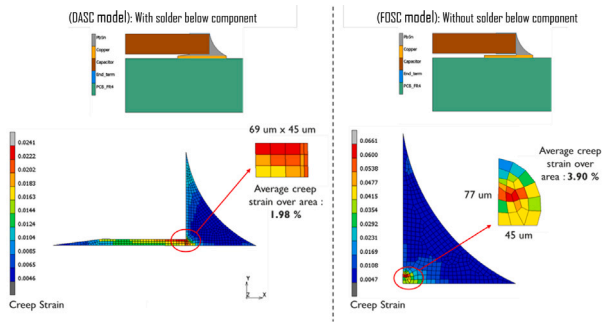


Fig. 4. FE analysis results showing average creep strain for a thermal cycle of −55 °C to 100 °C.

Fig. 5. Correlation matrix of input features to outputs of DASC model and FOSC model.

relationships between variables in a dataset [34]. Correlation values close to 1 or −1 indicate a strong relationship, while values near 0 suggest a weak or no relationship. Positive values mean both variables increase together, while negative values show one rises as the other falls. Fig. 5 shows the correlation between the input feature and the average creep strain in DASC model and FOSC model. It shows that the length of capacitor (lc) and CTE of PCB (PCB_CTE) positively impact both targets. Solder stand-off height (tsf) is a strong negative predictor, especially for FOSC model. The capacitor's coefficient of thermal expansion (Cap_CTE) also has a negative effect on both models, though to a lesser extent. Tmin show s a weaker correlation with both models, especially FOSC model, indicating lower minimum temperatures lead to lower average creep strain. Tmax has a weak positive correlation with both models, suggesting higher maximum temperatures slightly increase average strain. Features like capacitor width (wc), Partial copper pad Length (lsf1 and lsf2) might be less important due to weak correlations.

3.3.2. Min-max scaler

The Min-Max Scaler is a data pre-processing technique that scales each feature to a given range, usually [0, 1]. It adjusts each value based on the feature's minimum and maximum using the formula [35]:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

where X denotes the original value, X_{min} and X_{max} are the minimum and maximum of the feature column, X_{scaled} are the scaled value between 0 and 1. In this work, the Min-Max Scaler was applied to normalize the input features.

Following data normalization, the dataset was split into training and test subsets. For each simulation group, a consistent test set comprising 113 data points was employed to ensure comparability. The training set was used for hyperparameter optimization through Optuna (referred

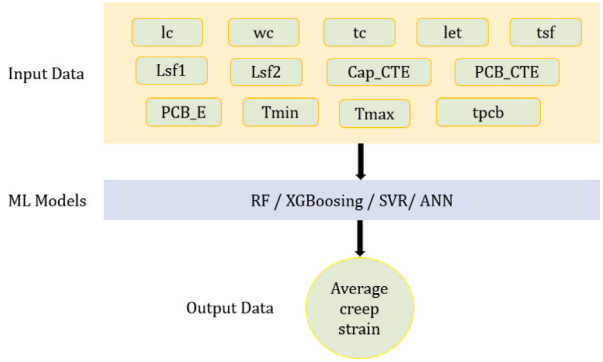


Fig. 6. Architecture of the modeling procedure.

Table 2
Hyperparameters of the RF model tuned in this study [36–39].

Hyperparameter	Description	Search range
`n_estimators`	Number of trees in the forest.	[10, 200]
`max_depth`	Maximum depth of each tree.	[50, 100]
`min_samples_split`	Minimum number of samples to split an internal node.	[2, 10]
`min_samples_leaf`	Minimum number of samples required at a leaf node.	[1, 15]
`max_features`	Method to select features at each split.	['sqrt', 'log2', None]

to Section 3.4.2), resulting in an optimized model trained on the same training set. Finally, the performance of this trained model was evaluated on the test set to facilitate comparison with other models.

3.4. Machine learning models

The ML models used in this study are trained on labeled datasets, where input parameters (input data) and corresponding output values (output data) are provided. This approach, known as supervised learning, encompasses techniques such as Random Forest (RF), eXtreme Gradient-boosting (XGB), Support Vector Regression SVR and Artificial Neural Network (ANN). Fig. 6 illustrates the schematic architecture of the ML models employed for predicting average creep strain.

3.4.1. Evaluated machine learning models

The first technique employed in this study is the random forest (RF). RF is an ensemble learning method that consists of several tree predictors where each tree is generated using a random vector sampled independently from the input vector [40]. In regression analysis, RF model computes the average predictions from multiple decision trees to generate the output [41]. Consequently, as the number of trees increases, the generalization error approaches a stable limit [42]. Table 2 indicates the hyperparameters and search ranges of the decision tree model used in this study [36–39]. As shown in Table 2, hyperparameters contain the number of trees in the forest (n_estimators), the max number of levels in each decision tree (max_depth), and the number of data points placed in a node before the node is split (min_samples_split).

RF is a basic tree-based machine learning algorithm, while eXtreme Gradient-boosting (XGB) is an advanced model of gradient-boosting decision trees [43]. XGB combines two key techniques for ensemble learning: bagging and boosting. Bagging trains multiple models in parallel, with each model generated from independently sampled subsets of the data. This approach reduces variance and improves model stability and accuracy by combining the predictions of all models. Boosting builds models sequentially, where each tree is constructed to address the errors of the previous one. This iterative process refines the model by focusing on the poorly learned patterns, thereby enhancing overall predictive performance [44,45]. Table 3 presents the hyperparameters and the search ranges of the XGB model in this study [46,47].

Table 3
Hyperparameters of the XGB model tuned in this study [46,47].

Hyperparameter	Description	Search range
`n_estimators`	Number of trees in the ensemble.	[30, 400]
`max_depth`	Maximum depth of each tree.	[3, 12]
`learning_rate`	Step size shrinkage during training.	[0.01, 0.1]
`subsample`	Subsample ratio of training instances.	[0.6, 0.9]
`colsample_bytree`	Subsample ratio of columns per tree.	[0.6, 0.9]
`reg_alpha`	L1 regularization term on weights.	[0.01, 10]
`reg_lambda`	L2 regularization term on weights.	[0.01, 10]

Table 4
Hyperparameters of the SVR model tuned in this study [48].

Hyperparameter	Description	Search range
`kernel`	Kernel function used to map input data to a higher-dimensional space.	['linear', 'poly', 'rbf', 'sigmoid']
`C`	Regularization parameter controlling the trade-off between smoothness and accuracy.	[1e−3, 50]
`epsilon`	Tolerance margin for error in the insensitive loss function.	[1e−7, 1e−3]
`gamma`	Defines the influence of a single training example in non-linear kernels.	['scale', 'auto']

Support vector regression (SVR) is a supervised machine learning algorithm that uses the principles of Support Vector Machines (SVM) to perform regression tasks by finding a function that predicts continuous values while maintaining a margin of tolerance around the predicted value and minimizing model complexity [49]. It identifies support vectors (data points closest to the margin) to define the decision boundary, enabling robust predictions even in high-dimensional spaces or with non-linear relationships when combined with kernel functions [50]. Table 4 illustrates hyperparameters and the search ranges of the SVR model in this study [48].

An Artificial Neural Network (ANN) is a machine learning model inspired by the human brain, designed to learn and map complex relationships in data [51]. It consists of an input layer that receives data, hidden layers where neurons compute weighted sums of inputs, add biases, and apply activation functions to capture non-linear patterns, and an output layer that produces predictions. During training, the network minimizes the error between predictions and actual values using backpropagation, where gradients are computed to update weights and biases iteratively through an optimization algorithm [51–53]. This iterative process enables ANNs to learn from data, making them highly effective for tasks like image recognition, natural language processing, and predictive analytics [54]. Table 5 illustrates hyperparameters and the search ranges of the ANN model in this study [55,56].

3.4.2. Optuna for selecting hyper-parameters of machine learning models

Hyperparameter tuning is essential for getting the best performance from machine learning models, as effective tuning can greatly improve both accuracy and robustness [19,57]. However, traditional tuning methods (Grid Search and Random Search) often require users to define a static search space in advance, which limits flexibility [58]. Optuna, proposed by Akiba et al. [59], is a hyperparameter optimization framework designed to address these limitations through a dynamic "define-by-run" application programming interface (API). This API allows users to build search spaces interactively during the optimization process [59,

Fig. 7. The basic procedure for hyperparameters selection in Optuna.

Table 5
Hyperparameters of the ANN model tuned in this study [55,56].

Hyperparameter	Description	Search range
num_layers	Number of hidden layers in the network.	3
hidden_dim1	Number of neurons in the first hidden layer.	[16, 256]
hidden_dim2	Number of neurons in the second hidden layer.	[32, 128]
hidden_dim3	Number of neurons in the third hidden layer.	[16, 128]
dropout_rate	Dropout rate to prevent overfitting.	[0.0, 0.4]
batch_norm	Whether to apply batch normalization after each layer.	[True, False]
learning_rate	Learning rate controlling the update step in training.	[1e−5, 1e−2]
weight_decay	Regularization to penalize large weights.	[1e−6, 1e−3]
epochs	Number of training epochs.	[50, 200]

60]. Meanwhile, it features efficient sampling and pruning algorithms with support for user customization. Sampling involves two types: independent (e.g., Tree-structured Parzen Estimator) and relational (e.g., Covariance Matrix Adaptation Evolution Strategy), enabling flexible and dynamic search space exploration. Pruning saves resources by monitoring intermediate results and terminating unpromising trials early, with Optuna implementing the state-of-the-art Asynchronous Successive Halving Algorithm (ASHA) for asynchronous and scalable pruning. By combining efficient sampling to identify promising areas and pruning to focus resources, Optuna optimizes performance while reducing computational overhead [59]. Optuna's final design feature is its easy set-up, making it simple to use for everything from small experiments to large-scale distributed tasks, supported by its flexible and adaptable architecture [59,61,62].

Fig. 7 illustrates the key steps involved in hyperparameter optimization for machine learning models using Optuna. The process starts with selecting the hyperparameters for each model and defining their respective search spaces. Next, the objective function is configured, followed by specifying the optimization direction. Finally, the number of Optuna trials is determined. In this study, hyperparameter optimization is performed over 100 trials with the objective of minimizing the MSE, evaluated through 5-fold cross-validation. This approach helps mitigate overfitting and ensures model robustness and generalizability.

3.5. Evaluation metrics

To assess the prediction accuracy of different machine learning models, three statistical metrics are employed: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the Coefficient of Determination (R^2). These metrics provide complementary insights into model performance, from error magnitude to variance explanation.

The Mean Squared Error (MSE) is defined as the mean of the overall squared prediction errors [63]. When the MSE is zero, the estimator precisely predicts the parameter's response. As the MSE decreases, the model's accuracy improves, reducing the discrepancy between predicted and actual values. It is defined as:

$$MSE = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} (y_i - \hat{y}_i)^2$$

, where y_i represents the actual value, $\hat{y}_i$ denotes the predicted value obtained from the constructed surrogate model and N_{test} indicates the total number of verification samples. In this study, identification of optimal model parameters using Optuna is conducted by minimizing the MSE of cross validation data.

The Root Mean Squared Error (RMSE) is derived from the MSE and provides a more easily interpretable measure of prediction error, due to having the same units as the target variable. It reflects the standard deviation of the residuals and is useful for understanding the typical magnitude of prediction errors. A lower RMSE suggests a better fit of the model to the data [63], and it is used here to compare the performance of the four machine learning models under consideration. It is defined as

$$RMSE = \sqrt{\frac{1}{N_{test}} \sum_{i=1}^{N_{test}} (y_i - \hat{y}_i)^2}$$

The Coefficient of Determination (R^2) evaluates how well the predicted values approximate the actual data by quantifying the proportion of variance in the target variable that is explained by the model [64]. R^2 takes values within the range [0, 1], with higher values indicating better predictive performance. In this work, R^2 is used alongside RMSE to compare the effectiveness of the different models, providing insight into how well each model captures the underlying patterns in the data. It is mathematically defined as

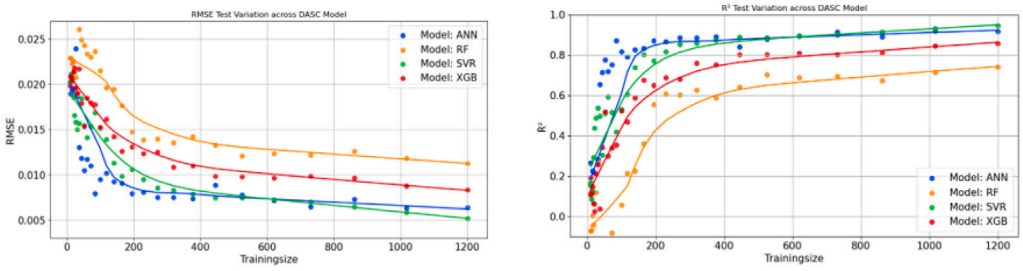
$$R^2 = \frac{\sum_{i=1}^{N_{test}} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{N_{test}} (y_i - \bar{y}_i)^2}$$

, with the term $\bar{y}_i$ referring to the mean of the true values [65].

4. Results and discussions

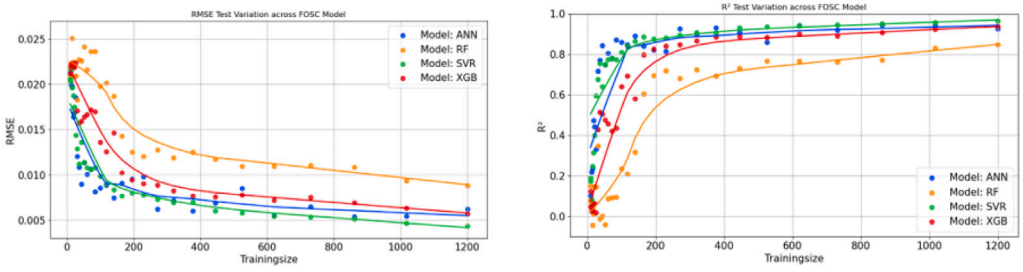
4.1. Model performance comparison

After hyperparameter optimization with Optuna, the optimal parameters for each model were identified, and models configured with these optimal parameters were subsequently applied to the test set to evaluate their performance. Four regression models were assessed and compared based on their performance on DASC model, using RMSE, and R^2 as evaluation metrics. The results, including fitted locally weighted scatterplot smoothing trends, are presented in Fig. 8. As the number of simulations increases, all models generally improve, with lower prediction errors (RMSE) and higher accuracy (R^2). The results show that ANN performs best in the small sample regime, with a rapid increase in R^2 and a steep decrease in RMSE, indicating strong nonlinear modeling capacity and good generalization. As the training size increases, SVR demonstrates robust and steadily improving performance, ultimately slightly outperforming ANN in the larger datasiye regime, making it one of the most stable models overall. XGBoost shows moderate and consistent performance throughout, while RF underperforms significantly, especially with small datasets, and continues to fall behind other models even as more data is introduced. Notably, all models exhibit diminishing returns in performance improvement as the training data size increases. Specifically, when the number of training samples reaches approximately 700, the R^2 values for ANN and SVR plateau and RMSE reduction slows, indicating that further improvements are limited. Therefore, focusing resources on ANN or



(a) RMSE as evaluation metric. (b) R^2 as evaluation metric.

Fig. 8. Comparison of 4 models on test set, for DASC model.



(a) RMSE as evaluation metric. (b) R^2 as evaluation metric.

Fig. 9. Comparison of 4 models on test set, for FOSC model.

SVR and maintaining simulation counts around or slightly above 700 would provide optimal balance and efficiency.

Fig. 9 shows the same comparison of FOSC model with DASC model. The results indicate that SVR consistently outperforms other models, achieving the highest R^2 and the lowest RMSE across most training sizes, especially in the large-data regime. ANN performs competitively, particularly excelling in small to medium training sizes, and remains close to SVR as the data size grows. XGBoost shows moderate performance, improving steadily with more data but falling short of SVR and ANN. Random Forest performs the worst, especially in low-data scenarios, and exhibits limited improvement even with increased data. A clear stabilization point is also observed: SVR and ANN reach performance saturation around 500–600 training samples, with marginal gains beyond that point. XGBoost stabilizes closer to 700–800 samples, while Random Forest shows no distinct saturation trend. This suggests that, for the FOSC task, a training size of approximately 600 samples is sufficient for achieving near-optimal model performance.

An important observation across all models is the phenomenon of diminishing returns. While all models benefit from an initial increase in training data (especially from 10 to 400 samples), the fitted curves show that beyond approximately 600 samples, the performance gains begin to plateau. The slope of the performance curves becomes increasingly shallow, indicating that additional data points contribute less and less to improving model accuracy. This trend is particularly evident in ANN and SVR, whose performance metrics approach their asymptotic limits. It means that simply increasing the dataset size does not indefinitely lead to significant improvements in performance [66].

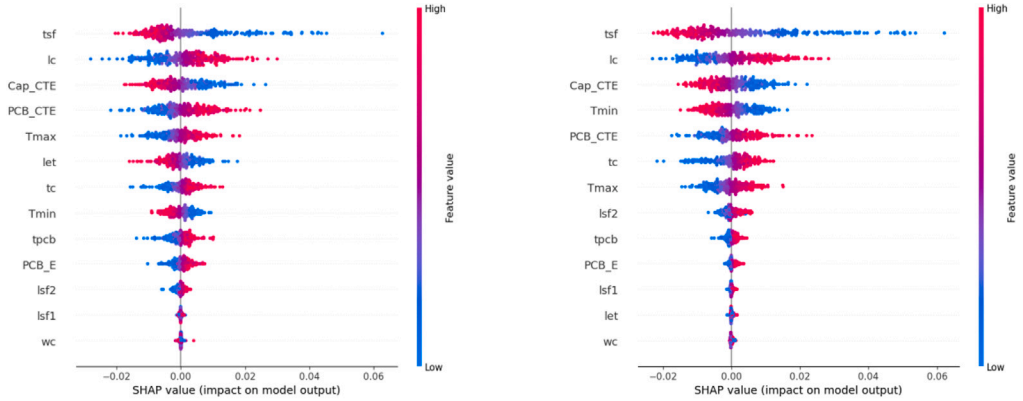
Numerically in Figs. 8 and 9, SVR consistently achieves the best performance across both models, with RMSE near 0.004 and R^2 consistently around or exceeding 0.96, especially at higher simulation counts. ANN closely follows SVR, performing strongly with the but slightly behind with RMSE near 0.005 and R^2 about 0.94. All models were trained on a local machine equipped with an Intel® Core™ i7 CPU and 32 GB RAM, using CPU-only computation. Under this setup (dataset size = 1200), the average optimization time for 100 trials was approximately 2 min for RF, 9 min for SVR, 3 min for ANN, and 1 min for XGBoost.

4.2. Interpretation of results

4.2.1. SHAP analysis in ANN

The SHapley Additive exPlanations (SHAP) framework is considered to explain the machine learning models [67,68]. It assigns each feature an importance value, called a SHAP value, based on the concept of Shapley values from cooperative game theory [68,69]. A SHAP value indicates how much each input feature contributes, positively or negatively, to creep strain predictions. In this study, SHAP values were computed using the KernelExplainer from the SHAP Python library, which is model-agnostic and well-suited for explaining non-tree-based models such as the ANN used here [70]. The explainer estimates feature contributions by perturbing input values and observing the corresponding changes in the model's output. The magnitude of a SHAP value reflects the strength of the feature's influence on the prediction, while the sign indicates whether the feature increases (positive) or decreases (negative) the predicted output. In summary plots, the color represents the actual value of the feature, providing insight into how high or low feature values affect the model's output.

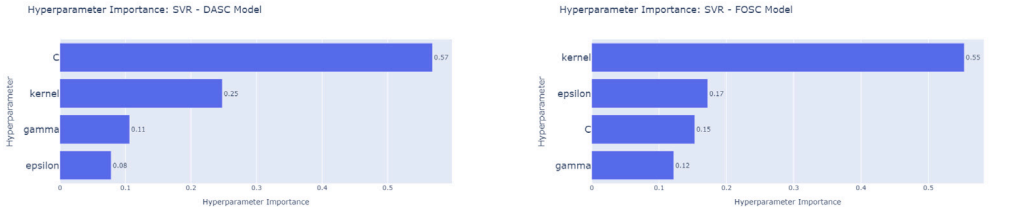
Fig. 10 depicts the SHAP analysis of ANN on both models. In both models, solder stand-off height (tsf) and component length (lc) are consistently the most influential predictors, showing wider distributions of SHAP values. Higher values of solder stand-off height (tsf) generally increase predictions significantly in both models. Features such as CTE of capacitors (Cap_CTE) and CTE of PCB (PCB_CTE) maintain relatively high importance in both models, reflecting stable influence. Specially in DASC model, Max temperature (Tmax) and end termination length (let) are relatively more influential, indicating that peak temperatures and horizontal structure are more critical for solder cracking below component (in the solder standoff area). In FOSC model, minimum temperature (Tmin) becomes notably more influential than maximum temperature (Tmax), suggesting that low temperature effects (e.g., contraction, stress recovery) play a larger role after cracking. Partial copper pad length at filler area (lsf2) gains significance, consistent with the focus shifting to the outer solder region. End termination length (let) in FOSC model becomes less important, indicating reduced influence as FOSC model assumes full fracture at solder below component. Overall, PCB thickness (tpcb), PCB young's modulus (PCB_E), partial copper



(a) DASC model: The features tsf, lc, and Cap_CTE have the highest impact on the model's output, with tsf being the most influential.

(b) FOSC model: Similar to DASC model, tsf and lc dominate in importance, but Tmin also plays a larger role compared to other features.

Fig. 10. SHAP values of each feature in the ANN model.



(a) DASC model: the most influential hyperparameter is regularization factor C.

(b) FOSC model: the most impact factor is choice of kernel function.

Fig. 11. Comparison of SVR performance across two datasets: DASC and FOSC.

pad length under component (lsf1), component width (wc) are lower ranked and have little influence on the solder joint creep strain in our current loading scenarios. This result is consistent with that of correlation matrix in Section 3.3.1.

4.2.2. *Hyperparameter importance analysis in SVR and ANN*

Optuna's hyperparameter importance feature helps to identify which hyperparameters have the most impact on model performance [59]. It analyzes completed trials and calculates each parameter's influence on the objective value. It helps save time and resources by focusing optimization on important parameters. Fig. 11 present the hyperparameter importance for SVR models trained on the DASC and FOSC datasets, respectively. In the DASC model, the most influential hyperparameter is C (0.57), indicating that regularization strength significantly affects model performance. The kernel parameter also plays a moderate role (0.25), while gamma and epsilon have relatively minor impacts (0.11 and 0.08, respectively). In contrast, the FOSC model is most sensitive to the kernel choice (0.55), making it the key factor in tuning. epsilon and C follow with moderate importance (0.17 and 0.15), and gamma remains the least impactful (0.12).

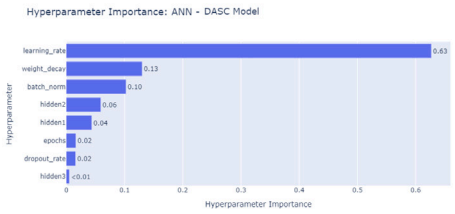
Fig. 12 shows a detailed analysis comparing the ANN hyperparameter importance for DASC model and FOSC model. In both, learning rate is the most influential parameter—especially in DASC Model (0.63 vs. 0.39). DASC model is also moderately affected by weight decay and batch normalization, while architectural parameters have little impact. In contrast, FASC model is more influenced by batch normalization, number of neurons in hidden layer 3, and dropout rate, indicating greater sensitivity to architecture and regularization.

Through SHAP analysis, we can identify the most influential design features affecting model performance. In this case, solder stand-off height (tsf) and the capacitor length (lc) are the most important features, thus avoiding unnecessary experimental designs in future studies. Additionally, hyperparameter importance analysis reveals which hyperparameters significantly impact model performance in SVR and ANN. SVR models show model-dependent sensitivity to either the regularization parameter C or the kernel function, while ANN models are consistently driven by the learning rate, with FOSC additionally influenced by architectural factors like batch normalization, enabling a more efficient optimization process.

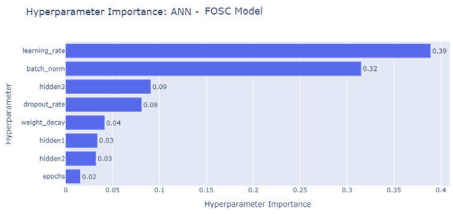
4.3. *Limitations and future work*

A key limitation of this study is the absence of experimental validation. The proposed framework relies exclusively on high-fidelity finite element analysis data to train and test the machine learning models. This synthetic data approach enables efficient exploration of parameter spaces and facilitates the construction of accurate and computationally efficient surrogate models. However, it inherently lacks direct correlation with physical test results. Although simulation data provide a consistent and controlled environment for model development, they may not fully capture the variability and complexity of real-world thermal cycling behavior.

To mitigate this concern, we emphasize that the FEA modeling approach adopted in this study builds upon well-established simulation methodologies that have previously demonstrated excellent agreement with experimental observations in similar contexts [71–74]. Nonetheless, we fully acknowledge that simulation-based predictions, regardless of their fidelity, cannot substitute for experimental validation. Critical factors such as material variability, process-induced defects, and environmental uncertainties are difficult to capture fully in numerical models alone. As such, experimental correlation remains an essential next step. In future work, we aim to collaborate with industrial partners to obtain accelerated thermal cycling test data, which will enable direct comparison and validation of the surrogate model's predictive capability under real-world operating conditions.



(a) DASC model: The learning rate overwhelmingly dominates in importance, while the influence of other hyperparameters is minimal.



(b) FOSC model: The learning rate is the most impactful hyperparameter, followed by batch normalization, both contributing significantly more than the others.

Fig. 12. Comparison of ANN performance across two datasets: DASC and FOSC.

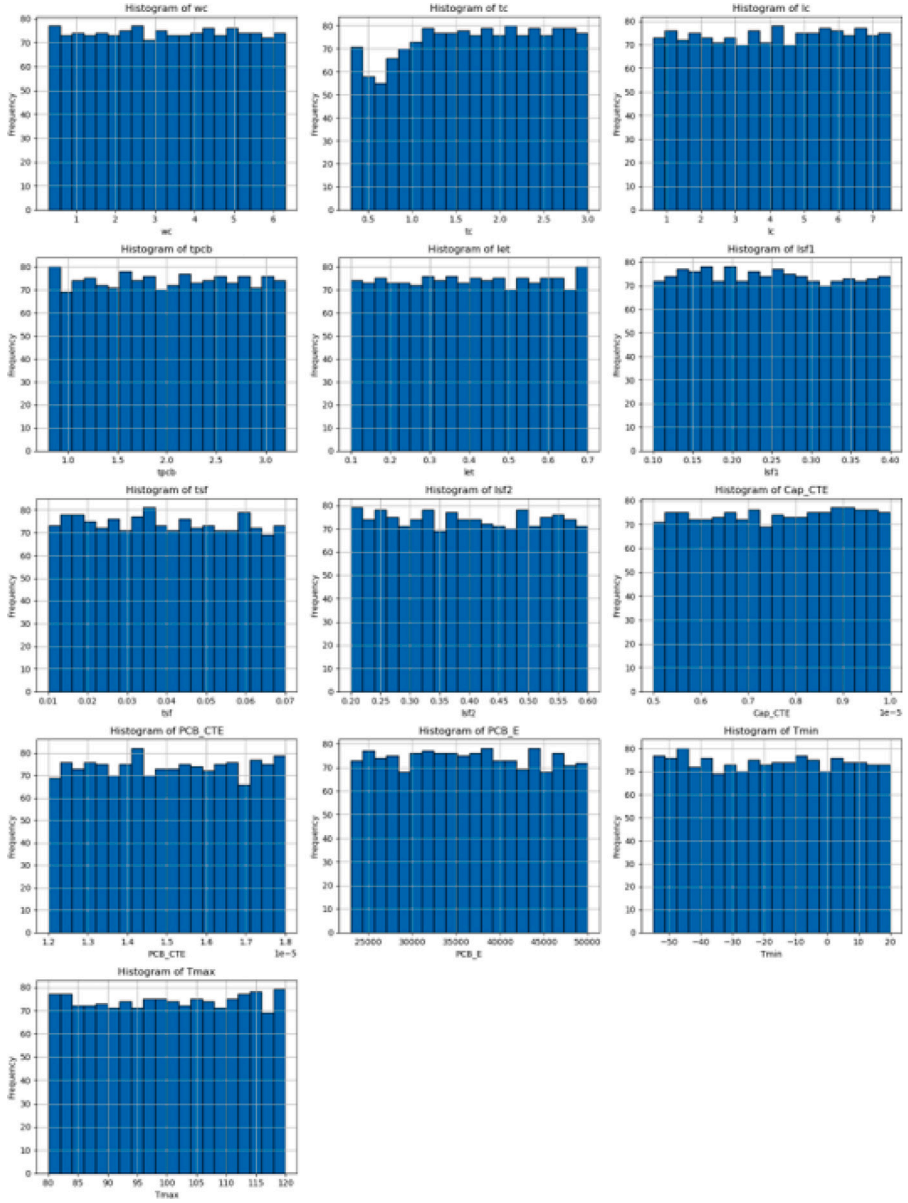


Fig. A.13. Histogram plots of input parameter distributions.

Another limitation of this study is the scope of applicability. The current framework is validated for MLCCs under thermal cycling conditions. Its extension to other package types, such as BGAs or QFNs, poses considerable challenges due to fundamental differences in package geometry, solder joint morphology, and material characteristics [31]. These factors significantly alter the thermo-mechanical behavior of the assembly, thereby requiring specialized modeling strategies for each package type. As such, the current framework is not intended to be universally applicable, and future work may focus on developing tailored models for other package families.

5. Conclusions

This study employed four advanced machine learning approaches. Each method was integrated with Optuna for hyperparameter optimization. The objective was to predict the average creep strain of solder joints in multilayer chip capacitors. Predictions were made under a range of simulation scenarios, which included variations in material properties, geometric configurations and thermal loading conditions. Two distinct finite element models were investigated: DASC model assumed crack propagation beneath the component area, whereas FOSC model presumed crack propagation within the solder fillet area.

The prediction results indicated that the four evaluated machine-learning models, optimized with Optuna, demonstrated feasibility and high accuracy for forecasting target creep strain values. Among these, the Support Vector Regression and Artificial Neural Network models exhibited superior predictive performance. Additionally, it was observed that the accuracy of the predictions improved significantly with increasing dataset size; however, after approximately 600–700 simulations, the rate of accuracy enhancement notably decreased.

In the ANN feature importance analysis, solder stand-off height and component length consistently emerged as the most influential parameters affecting prediction accuracy. Furthermore, the hyperparameter optimization analysis identified the learning rate as the most influential parameter for the accuracy of ANN model prediction, while the regularization parameter C or the kernel function plays the most critical role in for SVR model predictions.

The demonstrated predictive capabilities of machine learning models, particularly the Support Vector Regression and Artificial Neural Network optimized via Optuna, underscore their potential for advancing reliability assessments of multilayer chip capacitors. Future research should focus on expanding these models to enable full lifetime prediction by incorporating time-dependent material degradation and cyclic thermal-mechanical loading conditions. By incorporating degradation histories and failure criteria, the predictive models could evolve from single-point strain forecasts to comprehensive lifetime estimations. Such developments would enable more efficient reliability assessments and provide a foundation for proactive design and material selection strategies in electronic packaging.

CRediT authorship contribution statement

Qiulin Yu: Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing. **Chinmay Nawghane:** Data curation, Software, Validation, Writing – review & editing. **Zihan Zhang:** Investigation, Writing – review & editing. **Bart Vandevelde:** Conceptualization, Supervision. **Karl Fendt:** Supervision, Software, Writing – review & editing, Validation. **Thomas Krivec:** Supervision, Writing – review & editing. **Dieter P. Gruber:** Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Funded by the European Union (Grant Agreement No. 101072491). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

Appendix. Histogram analysis of input parameters

See Fig. A.13.

Data availability

The data that has been used is confidential.

References

[1] S. Yoshioka, S. Tani, M. Kumasawa, A. Inoue, Low cycle fatigue properties of solder material(36 Pb 62 Sn 2 Ag) at low temperatures, J. Soc. Mater. Sci. Japan 39 (442) (1990) 908–913.
[2] C. Kanchanomai, Y. Miyashita, Y. Mutoh, Strain-rate effects on low cycle fatigue mechanism of eutectic Sn–Pb solder, Int. J. Fatigue 24 (9) (2002) 987–993.
[3] J. Schijve, Significance of fatigue cracks in micro-range and macro-range, in: Fatigue Crack Propagation, ASTM International, 1967.
[4] S. De-Guang, Y. Wei-Xing, W. De-Jun, A new approach to the determination of fatigue crack initiation size, Int. J. Fatigue 20 (9) (1998) 683–687.
[5] K. Miller, The behaviour of short fatigue cracks and their initiation part ii-a general summary, Fatigue Fract. Eng. Mater. Struct. 10 (2) (1987) 93–113.
[6] J.H. Lau, Electronics manufacturing with lead-free, Halogen Free. Conduct. Adhesive Mater. 17 (2003).
[7] A. Dasgupta, C. Oyan, D. Barker, M. Pecht, Solder creep-fatigue analysis by an energy-partitioning approach, NA (1992).
[8] J.C. Suhling, H. Gale, R.W. Johnson, M.N. Islam, T. Shete, P. Lall, M.J. Bozack, J.L. Evans, P. Seto, T. Gupta, et al., Thermal cycling reliability of lead free solders for automotive applications, in: The Ninth Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (IEEE Cat. No. 04CH37543), Vol. 2, IEEE, 2004, pp. 350–357.
[9] J. Standard, Temperature Cycling, JESD22-A104D, JEDEC Solid State Technology Association, Arlington, 2009.
[10] X.-J. Fan, H. Wang, T. Lim, Investigation of the underfill delamination and cracking in flip-chip modules under temperature cyclic loading, IEEE Trans. Compon. Packag. Technol. 24 (1) (2001) 84–91.
[11] X. Fan, M. Pei, P.K. Bhatti, Effect of finite element modeling techniques on solder joint fatigue life prediction of flip-chip BGA packages, in: 56th Electronic Components and Technology Conference 2006, IEEE, 2006, 9pp.-.
[12] X. Fan, B. Varia, Q. Han, Design and optimization of thermo-mechanical reliability in wafer level packaging, Microelectron. Reliab. 50 (4) (2010) 536–546.
[13] S. Liu, S. Panigrahy, K. Chiang, Prediction of fan-out panel level warpage using neural network model with edge detection enhancement, in: 2020 IEEE 70th Electronic Components and Technology Conference, ECTC, IEEE, 2020, pp. 1626–1631.
[14] I.-R. Chen, F.B. Bastani, Effect of artificial-intelligence planning-procedures on system reliability, IEEE Trans. Reliab. 40 (3) (1991) 364–369.
[15] S.K. Panigrahy, Y.-C. Tseng, B.-R. Lai, K.-N. Chiang, An overview of AI-assisted design-on-simulation technology for reliability life prediction of advanced packaging, Mater. 14 (18) (2021) 5342.
[16] X. Zhao, H. Zheng, Z. Zhao, M. Cheng, W. Li, G. Wan, Y. Jia, Warpage prediction of fan-out wafer-level package based on coupled deep learning and finite element simulation, Microelectron. Reliab. 170 (2025) 115759.
[17] M. Sperti, C. Nawghane, B. Vandevelde, N. Lammens, M. Verbeke, Physics-informed machine learning-based methodology for plated through holes lifetime estimation in printed circuit boards, in: 2025 26th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems, EuroSimE, IEEE, 2025, pp. 1–10.
[18] H. Hsiao, K. Chiang, AI-assisted reliability life prediction model for wafer-level packaging using the random forest method, J. Mech. 37 (2020) 28–36.
[19] L. Yang, A. Shami, On hyperparameter optimization of machine learning algorithms: Theory and practice, Neurocomputing 415 (2020) 295–316.
[20] A. Syed, Accumulated creep strain and energy density based thermal fatigue life prediction models for SnAgCu solder joints, in: 2004 Proceedings. 54th Electronic Components and Technology Conference (IEEE Cat. No. 04CH37546), Vol. 1, IEEE, 2004, pp. 737–746.

[21] J. Lau, W. Danksher, P. Vianco, Acceleration models, constitutive equations, and reliability of lead-free solders and joints, in: 53rd Electronic Components and Technology Conference, 2003. Proceedings, IEEE, 2003, pp. 229–236.
[22] A. Schubert, R. Dudek, E. Auerswald, A. Gollhardt, B. Michel, H. Reichl, Fatigue life models for SnAgCu and SnPb solder joints evaluated by experiments and simulation, in: Electronic Components and Technology Conference, IEEE; 1999, 2003, pp. 603–610.
[23] J.H. Pang, B. Xiong, T. Low, Low cycle fatigue models for lead-free solders, Thin Solid Films 462 (2004) 408–412.
[24] C.-M. Liu, K.-N. Chiang, Solder shape design and thermal stress/strain analysis of flip chip packaging using hybrid method, in: International Symposium on Electronic Materials and Packaging (Cat. No. 00EX458), EMAP2000, IEEE, 2000, pp. 44–50.
[25] C.-M. Liu, C.-C. Lee, K.-N. Chiang, Enhancing the reliability of wafer level packaging by using solder joints layout design, IEEE Trans. Compon. Packag. Technol. 29 (4) (2006) 877–885.
[26] L. Xu, Z.X. Han, R. Wei, B.P. Wang, T. Reinikainen, Design optimization approaches for the thermo-mechanical reliability of land grid array solder joints, in: Proceedings of 2004 International Conference on the Business of Electronic Product Reliability and Liability (IEEE Cat. No. 04EX809), IEEE, 2004, pp. 107–112.
[27] T. Dragičević, P. Wheeler, F. Blaabjerg, Artificial intelligence aided automated design for reliability of power electronic systems, IEEE Trans. Power Electron. 34 (8) (2018) 7161–7171.
[28] H. Wankerl, M.L. Stern, P. Altieri-Weimar, S. Al-Baddai, K.-J. Lang, F. Roider, E.W. Lang, Fully convolutional networks for void segmentation in X-ray images of solder joints, J. Manuf. Process. 57 (2020) 762–767.
[29] L. Benabou, Development of LSTM networks for predicting viscoplasticity with effects of deformation, strain rate, and temperature history, J. Appl. Mech. 88 (7) (2021) 071008.
[30] C.C. Yuan, C.-C. Lee, Solder joint reliability modeling by sequential artificial neural network for glass wafer level chip scale package, IEEE Access 8 (2020) 143494–143501.
[31] P. Ferrando-Villalba, B. Vandevelde, BGA solder strain prediction using an artificial neural network regressor, in: 2022 23rd International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems, EuroSimE, IEEE, 2022, pp. 1–7.
[32] Hexagon AB, MSC marc, version 2021.2, 2021, URL https://hexagon.com/products/marc.
[33] Noesis Solutions, Noesis optimus, version 2022.2, 2022, URL https://www.noesissolutions.com/our-products/optimus.
[34] J.O. Ramsay, J. ten Berge, G.P. Styan, Matrix correlation, Psychom. 49 (3) (1984) 403–423.
[35] H. Henderi, T. Wahyuningsih, E. Rahwanto, Comparison of Min-Max normalization and Z-score normalization in the K-nearest neighbor (kNN) algorithm to test the accuracy of types of breast cancer, Int. J. Inform. Inf. Syst. 4 (1) (2021) 13–20.
[36] G. Silva, B. Schulze, M. Ferro, Performance and energy efficiency analysis of machine learning algorithms towards green AI: a case study of decision tree algorithms, Natl. Lab. Sci. Comput. (2021).
[37] E.M. García, M.G. Alberti, A.A. Arcos Álvarez, Measurement-while-drilling based estimation of dynamic penetrometer values using decision trees and random forests, Appl. Sci. 12 (9) (2022) 4565.
[38] N.M. Khiem, Y. Takahashi, H. Yasuma, K.T.P. Dong, T.N. Hai, N. Kimura, A novel machine learning approach to predict the export price of seafood products based on competitive information: The case of the export of Vietnamese shrimp to the US market, PLoS One 17 (9) (2022) e0275290.
[39] W. Khrakhuean, P. Chutima, Real-time induction motor health index prediction in a petrochemical plant using machine learning, Eng. J. 26 (5) (2022) 91–107.
[40] B. Singh, P. Sihag, K. Singh, Modelling of impact of water quality on infiltration rate of soil by random forest regression, Model. Earth Syst. Environ. 3 (2017) 999–1004.
[41] G.-F. Fan, M. Yu, S.-Q. Dong, Y.-H. Yeh, W.-C. Hong, Forecasting short-term electricity load using hybrid support vector regression with grey catastrophe and random forest modeling, Util. Policy 73 (2021) 101294.
[42] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32.
[43] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.
[44] H.A. Ali, C. Mohamed, B. Abdelhamid, N. Ourdani, T. El Alami, A comparative evaluation use bagging and boosting ensemble classifiers, in: 2022 International Conference on Intelligent Systems and Computer Vision, ISCV, IEEE, 2022, pp. 1–6.
[45] M. Guang, C. Yan, G. Liu, J. Wang, C. Jiang, A novel neighborhood-weighted sampling method for imbalanced datasets, Chin. J. Electron. 31 (5) (2022) 969–979.
[46] J. Hancock, T.M. Khoshgoftaar, Performance of catboost and xgboost in medicare fraud detection, in: 2020 19th IEEE International Conference on Machine Learning and Applications, ICMLA, IEEE, 2020, pp. 572–579.
[47] Z. Huang, M. Yang, B. Yang, W. Liu, Z. Chen, Data-driven model for predicting production periods in the SAGD process, Petroleum 8 (3) (2022) 363–374.
[48] Y. Sun, S. Ding, Z. Zhang, W. Jia, An improved grid search algorithm to optimize SVR for prediction, Soft Comput. 25 (2021) 5633–5644.
[49] W.S. Noble, What is a support vector machine? Nature Biotechnol. 24 (12) (2006) 1565–1567.
[50] S.M. Clarke, J.H. Griebsch, T.W. Simpson, Analysis of support vector regression for approximation of complex engineering analyses, 2005.
[51] A. Dongare, R. Kharde, A.D. Kachare, et al., Introduction to artificial neural network, Int. J. Eng. Innov. Technol. (IJEIT) 2 (1) (2012) 189–194.
[52] B. Yegnanarayana, Artificial Neural Networks, PHI Learning Pvt. Ltd, 2009.
[53] S. Agatonovic-Kustrin, R. Beresford, Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research, J. Pharm. Biomed. Anal. 22 (5) (2000) 717–727.
[54] Y.-c. Wu, J.-w. Feng, Development and application of artificial neural network, Wirel. Pers. Commun. 102 (2018) 1645–1656.
[55] S.C. Smithson, G. Yang, W.J. Gross, B.H. Meyer, Neural networks designing neural networks: multi-objective hyper-parameter optimization, in: 2016 IEEE/ACM International Conference on Computer-Aided Design, ICCAD, IEEE, 2016, pp. 1–8.
[56] G.I. Diaz, A. Fokoue-Nkoutche, G. Nannicini, H. Samulowitz, An effective algorithm for hyperparameter optimization of neural networks, IBM J. Res. Dev. 61 (4/5) (2017) 9–1.
[57] R. Andonie, Hyperparameter optimization in learning systems, J. Membr. Comput. 1 (4) (2019) 279–291.
[58] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2) (2012).
[59] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019.
[60] J.-P. Lai, Y.-L. Lin, H.-C. Lin, C.-Y. Shih, Y.-P. Wang, P.-F. Pai, Tree-based machine learning models with optuna in predicting impedance values for circuit analysis, Micromach. 14 (2) (2023) 265.
[61] S. Chintakindi, A. Alsamhan, M.H. Abidi, M.P. Kumar, Annealing of monel 400 alloy using principal component analysis, hyper-parameter optimization, machine learning techniques, and multi-objective particle swarm optimization, Int. J. Comput. Intell. Syst. 15 (1) (2022) 18.
[62] M. Sipper, High per parameter: A large-scale study of hyperparameter tuning for machine learning algorithms, Algorithms 15 (9) (2022) 315.
[63] T. Chai, R.R. Draxler, et al., Root mean square error (RMSE) or mean absolute error (MAE), Geosci. Model. Dev. Discuss. 7 (1) (2014) 1525–1534.
[64] A. Di Bucchianico, Coefficient of determination (R 2), Encycl. Stat. Qual. Reliab. (2008).
[65] J.P. Barrett, The coefficient of determination—some limitations, Amer. Statist. 28 (1) (1974) 19–20.
[66] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, I. Sutskever, Deep double descent: Where bigger models and more data hurt, J. Stat. Mech. Theory Exp. 2021 (12) (2021) 124003.
[67] L.S. Shapley, et al., A value for n-person games, 1953.
[68] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Adv. Neural Inf. Process. Syst. 30 (2017).
[69] E. Mosca, F. Szigeti, S. Tragianni, D. Gallagher, G. Groh, SHAP-based explanation methods: a review for NLP interpretability, in: Proceedings of the 29th International Conference on Computational Linguistics, 2022, pp. 4593–4603.
[70] Z. Li, Extracting spatial effects from machine learning model using local interpretation method: An example of SHAP and XGBoost, Comput. Environ. Urban Syst. 96 (2022) 101845.
[71] C. Nawghane, T. Moncond'Huy, B. Vandevelde, P. Vernhes, R. Cruz, Deformation analysis of QFN packages for validation of thermo-mechanical finite element simulations, in: 2023 24th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems, EuroSimE, IEEE, 2023, pp. 1–8.
[72] J. Zündel, T. Krivec, M. Weninger, Life-time prediction of copper μ-vias based on a stochastic design variation approach, in: 2024 25th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems, EuroSimE, IEEE, 2024, pp. 1–7.
[73] B. Vandevelde, R. Labie, D. Vanderstraeten, E. Blansaer, Impact of PCB-housing-interaction on QFN solder joint reliability, in: 2020 21st International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems, EuroSimE, IEEE, 2020, pp. 1–3.
[74] C. Nawghane, B. Vandevelde, R. Labie, B. Allaert, R. Lauwaert, F. Vanhee, D. Pissoort, I. De Wolf, J. Mehner, Vibration fatigue analysis of lead-free CSP assemblies on printed circuit board, in: 2018 19th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems, EuroSimE, IEEE, 2018, pp. 1–8.