micromachines

Article

RLC Circuit Forecast in Analog IC Packaging and Testing by Machine Learning Techniques

Jung-Pin Lai [1], Ying-Lei Lin [1], Ho-Chuan Lin [2], Chih-Yuan Shih [2], Yu-Po Wang [2] and Ping-Feng Pai [1,3,*]

[1] PhD Program in Strategy and Development of Emerging Industries, National Chi Nan University, Nantou 54561, Taiwan
[2] Siliconware Precision Industries Co., Ltd., No. 123, Sec. 3, Dafeng Rd., Dafeng Vil., Tanzi Dist., Taichung City 42749, Taiwan
[3] Department of Information Management, National Chi Nan University, Nantou 54561, Taiwan
* Correspondence: paipf@ncnu.edu.tw

Abstract: For electronic products, printed circuit boards are employed to fix integrated circuits (ICs) and connect all ICs and electronic components. This allows for the smooth transmission of electronic signals among electronic components. Machine learning (ML) techniques are popular and employed in various fields. To capture the nonlinear data patterns and input–output electrical relationships of analog circuits, this study aims to employ ML techniques to improve operations from modeling to testing in the analog IC packaging and testing industry. The simulation calculation of the resistance, inductance, and capacitance of the pin count corresponding to the target electrical specification is a complex process. Tasks include converting a two-dimensional circuit into a three-dimensional one in simulation and modeling-buried structure operations. In this study, circuit datasets are employed for training the ML model to predict resistance (R), inductance (L), and capacitance (C). The least squares support vector regression (LSSVR) with Genetic Algorithms (GA) (LSSVR-GA) serves as an ML model for forecasting RLC values. Genetic algorithms are used to select parameters of LSSVR models. To demonstrate the performance of LSSVR models in forecasting RLC values, three other ML models with genetic algorithms, including backpropagation neural networks (BPNN-GA), random forest (RF-GA), and eXtreme gradient boosting (XGBoost-GA), were employed to cope with the same data. Numerical results illustrated that the LSSVR-GA outperformed the three other forecasting models by around 14.84% averagely in terms of mean absolute percentage error (MAPE), weighted absolute percent error measure (WAPE), and normalized mean absolute error (NMAE). This study collected data from an IC packaging and testing firm in Taiwan. The innovation and advantage of the proposed method is using a machine approach to forecast RLC values instead of through simulation ways, which generates accurate results. Numerical results revealed that the developed ML model is effective and efficient in RLC circuit forecasting for the analog IC packaging and testing industry.

Keywords: integrated circuit; packaging and testing; machine learning; LSSVR

check for updates

Citation: Lai, J.-P.; Lin, Y.-L.; Lin, H.-C.; Shih, C.-Y.; Wang, Y.-P.; Pai, P.-F. RLC Circuit Forecast in Analog IC Packaging and Testing by Machine Learning Techniques. *Micromachines* **2022**, *13*, 1305. https://doi.org/10.3390/mi13081305

Academic Editor: Piero Malcovati

Received: 29 June 2022
Accepted: 9 August 2022
Published: 12 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Circuit simulation work in the integrated circuit (IC) packaging process depends on model complexity in geometry and electromagnetic materials. Properly simulating IC packaging plays a vital role in catching potential EMC, power, and signal integrity issues early in the design process and overcoming major pitfalls. The elimination of manual processing is required to reduce the time and effort from customer requirement specifications to IC packaging design, testing, manufacturing, and to determine the tools and process options for application design. Implementation takes prediction, evaluation, and decision making with machine learning-centric databases, tools, and design models. Learning-based tools and process models must continuously improve through additional design experience [1,2].

Machine learning (ML) solved many problems that were originally difficult to solve in data science. Many studies have shown that machine learning and optimization algorithms are suitable for solving different problems in the IC packaging and design processes, reducing design errors and design cycle time [3]. Table 1 shows the latest research on the IC packaging process using machine learning.

The signal passes through the substrate of the printed circuit board. Simulation provides the designer with a pre-optimized design concept. Ren et al. [4] introduced a graph neural network to predict network parasitics and device parameters by converting circuit schematics into graphs and utilizing key GNN-based modeling techniques. The results showed that the average simulation error was reduced from over 100% and estimated by designers to be less than 10%. Shook et al. [5] proposed a new machine learning-based parasitic estimation method for pre-layout custom circuit designs. For various analog circuits, the results show a reduction in the average error between pre-layout and post-layout circuit simulations from 37% to 8%.

For optimization and evaluation of package structural characteristics, Wu and Chu [6] proposed and verified an analog-driven design method for chip package integration structure design optimization. The study's results suggest that the random forest algorithm can predict stress for chip package-integrated design. Hsiao and Chiang [7] proposed applying the RF model to predict the reliability of wafer-level packaging. The designers can easily optimize the WLP structure and shorten the design cycle. Lee et al. [8] developed a chip-to-package interactive risk assessment platform using finite element analysis, meta-modeling, and genetic algorithm optimization methods.

Heat transfer analysis of package structures is important in package functional testing. Acharya et al. [9] used three ML algorithms, random forest, support vector regression, and a neural network to model thermal behavior through hotspot temperature simulation data evaluation. They proposed an ML-based thermal design method and provided a reference frame for future packaging materials. Durgam et al. [10] used several machine learning methods to predict the temperature of the heat source on the substrate. The results showed that the temperature agreement between the prediction and the simulation was less than 10%. Jing et al. [11] proposed using the genetic algorithm to optimize the temperature curve prediction model in the reflow soldering process. The results show that the predicted value meets the error accuracy requirements. The results also prove that the established mathematical model can effectively predict temperature curves.

The power delivery network (PDN) must reliably supply power to functional blocks in an integrated circuit (IC). A robust PDN design has always been a critical challenge. Cecchetti et al. [12] developed a Genetic Algorithm (GA) and Artificial Neural Network (ANN) model for iterative optimization of the placement of decoupling capacitors in a PDN. They concluded that the GA-ANN model is consistent with the results of commercial simulator optimization. Sourav et al. [13] presented an ML architecture that combined neural networks and regression trees to predict printed circuit board (PCB) inductance and resistance. They employed an LSTM model to predict voltage drop as a function of time. The average prediction accuracy of the proposed method is 94%.

There have been many models utilizing machine learning and optimization to solve issues in IC packaging. Mao et al. [14] proposed a machine learning (ML) model based on the backpropagation (BP) method for predicting three-dimensional board-level drop responses for ball grid array (BGA) encapsulation structures. Jin et al. [15] constructed several machine learning methods to accurately predict the radiated electric field of wire-bonded ball grid array packages. They optimized model parameters to minimize the prediction error of each model. Their conclusion shows that DNN is an effective and feasible prediction model. Wang et al. [16] proposed a reverse design method based on convolutional neural networks for the fast optimization and design of encapsulation structures. Schierholz et al. [2] provided a database that allows for the study of machine learning tools and techniques in signal integrity, power integrity, and electromagnetic compatibility. It contains printed circuit board (PCB)-based interconnects and physics (PB).

The corresponding frequency domain data of the tool can be used for different types of structural simulations.

Table 1. Summary of ML-based IC packaging process applications.

Literature	Years	Application	Method(s)
Ren et al. [4]	2021	Predict net parasitics and device parameters	GNN
Shook et al. [5]	2020	Parasitic estimation	Random forest
Wu and Chu [6]	2021	The structural design optimization of chip package integration	Random forest
Hsiao and Chiang [7]	2020	Packaging reliability analysis and prediction	Random forest
Lee et al. [8]	2021	Interactive risk assessment of chip packaging	FEA, MOGA
Acharya et al. [9]	2021	Predict the thermal behavior of a power electronics package	Random forest, SVR, ANN
Durgam et al. [10]	2022	The optimization of temperature on printed circuit board	XG Boost, ANN, SVR, RFR
Jing et al. [11]	2021	Predicting the temperature curve of SMT reflow soldering	Genetic Algorithm
Cecchetti et al. [12]	2020	Power delivery network (PDN)	ANN, Genetic Algorithm
Sourav et al. [13]	2020	Power delivery network (PDN)	Regressor trees, LSTM
Mao et al. [14]	2022	Predicting three-dimensional board-level drop responses for ball grid array (BGA) encapsulation structures	BPNN
Jin et al. [15]	2022	Predicting the radiated electric field of a wire-bonded ball grid array package	DNN, SVR, K-nearest neighbors, LR
Wang et al. [16]	2021	Full wave radiation simulation of package design process	CNN
Schierholz et al. [2]	2021	Signal integrity (SI) and power integrity (PI) database based on PCB interconnection	ANN, Genetic Algorithm

This study attempted to use least squares support vector regression with genetic algorithms to predict the RLC (resistance, inductance, and capacitance) values currently generated by simulation methods. The genetic algorithms were employed to determine LSSVR parameters to improve forecasting accuracy. The designed method employs a machine learning approach to forecast RLC values instead of through simulation ways and generates more accurate results than the three other machine learning models. The rest of this study is organized as follows. Section 2 provides the substrate and interface electrical transfer properties based on IC packages. Section 3 briefs the LSSVR model and genetic algorithms. The flowchart of the LSSVR-GA model for predicting RLC values is also addressed and presented. Numerical results are illustrated in Section 4. Conclusions are indicated in Section 5.

2. The Substrate and Interface of the IC Package Transmit Electrical Properties

In IC packaging design, the substrate is used as a carrier. The functions of the substrate are to protect and carry the IC chip and serve as a medium for circuit signal transmission. Integrated circuit packaging is the final stage of semiconductor component manufacturing. As a method for connecting the die to the external circuit, the chip's packaging considers the pin configuration, electrical performance, heat dissipation, and the chip's physical size. There are many typical packaging forms in the semiconductor industry [17,18]. Currently, the most common internal packaging methods of integrated circuits are wire bonding (WB) and flip chip (FC) packages. Flip chip packaging connects the chip to the bump and then turns the IC chip over to directly connect the bump and substrate. The wire bonding package places the chip on the substrate (chip pad) and then uses the wire bonding technology to connect the chip to the connection point on the substrate. The IC substrate acts as a buffer interface for electrical connection and transmission between the

IC die and the PCB through the conductive routing and vias (VIA) network, as shown in Figure 1.

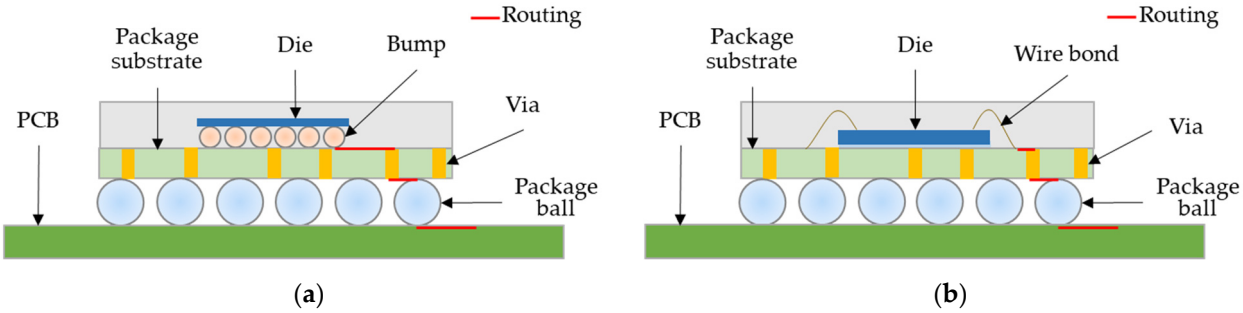


Figure 1. IC Package connections with substrate [19]. (**a**) Flip chip; (**b**) wire bonding.

The RLC circuit is essential to evaluate the overall interface transmission capability in the IC packaging design process. It is a circuit structure composed of resistors, capacitors, and inductors. Parasitic effects associated with ICs and printed circuit board (PCB) conductors and their paths are essential parameters of the electrical transport model. The parasitic effects of RLC lines in the IC package process can cause signal integrity problems due to signal attenuation and delay [20,21].

In the process of IC substrate generation, the substrate design is first performed according to the target circuit specification. The netlist of the corresponding circuit is associated with performing a post-layout simulation to verify the corresponding layout performance. If the post-layout simulation results are violated, the designer will adjust his layout and re-simulate. Figure 2 shows that this process is repeated until the simulated substrate design conforms to the RLC electrical specifications for interface transmission. The current process requires multiple simulation runs to meet the desired target circuit specification. Therefore, any inaccuracies in the design or components can produce misleading post-layout simulation results. Such misleading results can reduce yield and increase circuit design waste time.

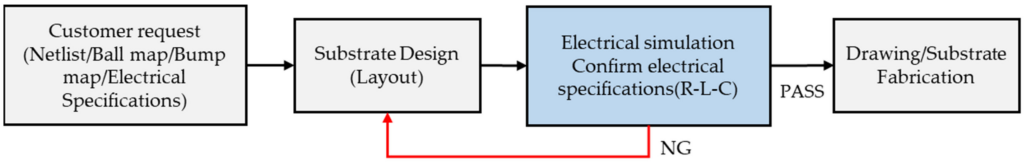


Figure 2. IC substrate generation process.

Signal transmission relies on the interconnected line group. According to the transmission line theory, the transmission line calibration model can replace the electrical characteristics of the signal and use an equivalent model. When the system simulation is based on the transmission line calibration model, the substrate RLC model, IC input/output buffer information specification (IBIS) model, and PCB electrical properties model are applied to the system simulations for system verification. The process is shown in Figure 3.

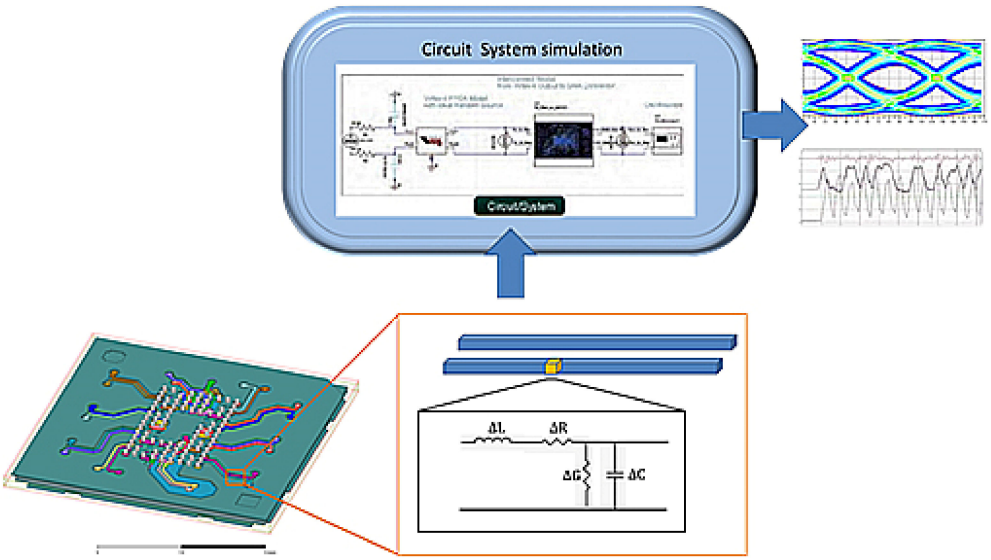


Figure 3. The illustration of RLC model system simulation and verification provided by SPIL (Siliconware Precision Industries Co., Ltd., Taichung City, Taiwan).

3. Forecasting RLC Values of Integrated Circuits by LSSVR-GA Models

3.1. LSSVR Models with Genetic Algorithms

The LSSVR method can be traced back to the SVM (Support Vector Machine) proposed by Cortes and Vapnik [22]. The SVM can handle classification and regression problems and performs better on small samples. Suykens and Vandewalle [23] proposed LSSVM. It solves the high computational burden problem of the SVM. The problem used to solve regression is called LSSVR [24].

Consider a given data set $\{x_i, y_i \mid i = 1,2,3, \ldots, n\}$, where $x_i \in R^d$ is the i^{th} input data including d features, and $y_t \in R$ is the i^{th} output data. Establishing the model for the LSSVR is as follows in Equation (1):

$$y\ (x) = \omega^T \cdot \varphi(x) + b \tag{1}$$

where ω^T is the transposed form of the weight matrix, $\varphi(x)$ represents a nonlinear function that maps from the original dimensional feature space to a higher dimensional feature space, and b is a bias value.

The optimization problem to be solved by the model is presented as Equation (2):

$$\begin{aligned} &Min\ F(\omega,\ e) = \tfrac{1}{2}\omega^T\omega + \tfrac{1}{2}\gamma \sum_{i=1}^{n} e_i^2 \\ &Subject\ to: y_i = \omega^T\varphi(x_i) + b + e_i \end{aligned} \tag{2}$$

where $F(\omega,\ b)$ is lose function, γ is the regularization parameter, and e means the random error.

Because of the constraints, the optimal solution to the computational problem is very complicated. The Lagrange function is optimized and presented in Equation (3) to solve this problem:

$$L(\omega,\ b,\ e_i,\ l_i) = F(\omega,\ b) - \sum_{i=1}^{n} l_i(\ \omega^T\varphi(x_i) + b + e_i - y_i) \tag{3}$$

where $L(\omega, b, e_t, l_t)$ is the Lagrange function and l is the Lagrange multiplier. After optimization using the KKT condition (Karush–Kuhn–Tucker condition), the formula is described in Equation (4)

$$\begin{gathered} \frac{\partial L}{\partial \omega} = 0 \to \omega = \sum_{i=1}^{n} l_i \varphi(x_i) \\ \frac{\partial L}{\partial b} = 0 \to \sum_{i=1}^{n} l_i = 0 \\ \frac{\partial L}{\partial e_i} = 0 \to l_i = \gamma e_i \\ \frac{\partial L}{\partial l_i} = 0 \to \omega^T \varphi(x_i) + b + e_i - y_i = 0 \end{gathered} \tag{4}$$

The kernel function $K(x, x_t)$ is considered as follows in Equation (5):

$$K(x, x_i) = \varphi(x)^T \varphi(x_i) \tag{5}$$

Finally, the model estimation formula by LSSVR can be obtained with Equation (6):

$$y = \sum_{i=1}^{n} l_i K(x, x_i) + b \tag{6}$$

There are common kernel functions such as the string kernel [25], the radial basis function kernel (RBF) [26], and the polynomial kernel [27]. This study used the RBF kernel function in Equation (7), and the RBF kernel utilizes high-dimensional nonlinear mapping to resolve the nonlinear relationship between dependent and independent variables. The RBF kernel learned more complex decision boundaries:

$$K(x, x_i) = \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right) \tag{7}$$

where σ is the parameter of the RBF kernel function. The decision of these two parameters—γ and σ—would affect the accuracy of the LSSVR model, so GA was performed to optimize these two parameters. The complete concept of GA was advocated by John Holland [28,29]. GA simulates the natural evolution law of natural ecology, imitates the survival of the fittest in the natural group, eliminates the inferior, and converges into a balanced mechanism under repeated iterations. GA is a search method used to solve optimization problems. Genes can select, crossover, and mutate. Better genes are passed to the new generation, and the inferior genes will be eliminated gradually. GA has been widely applied in solving optimization problems, data searches, artificial intelligence, and machine learning.

3.2. LSSVR-GA Architecture for RLC Prediction

Figure 4 illustrates the framework of the LSSVR-GA model in RLC (resistance, inductance, and capacitance) forecasting. It consists of 3 modules: data preprocessing, GA for parameter selection, and the LSSVR model for RLC forecasting. To solve the time-consuming problem of calculating the parameters for complex simulation software while verifying substrate designs, this study proposed a machine learning method to predict the RLC values for different product types.

The experimental data were semi-structured historical data provided by SPIL (Siliconware Precision Industries Co., Ltd.), including two different IC package process products, FC and WB. The historical data of different products of two-layer, four-layer, and six-layer PCB were selected based on these two processes. The three dependent variables for each product are resistance (R), inductance (L), and capacitance (C). The independent variables (X1~Xn) include ball, bump, base, L1, L2, L3, L4, L5, via, and wire. In the data preprocessing stage, this study integrated these scattered semi-structured data into one-to-one corresponding structured data between dependent and independent variables according to product types. It filled the missing values with 0. Table 2 describes the features and samples in predicting RLC for different data sets of substrate products.

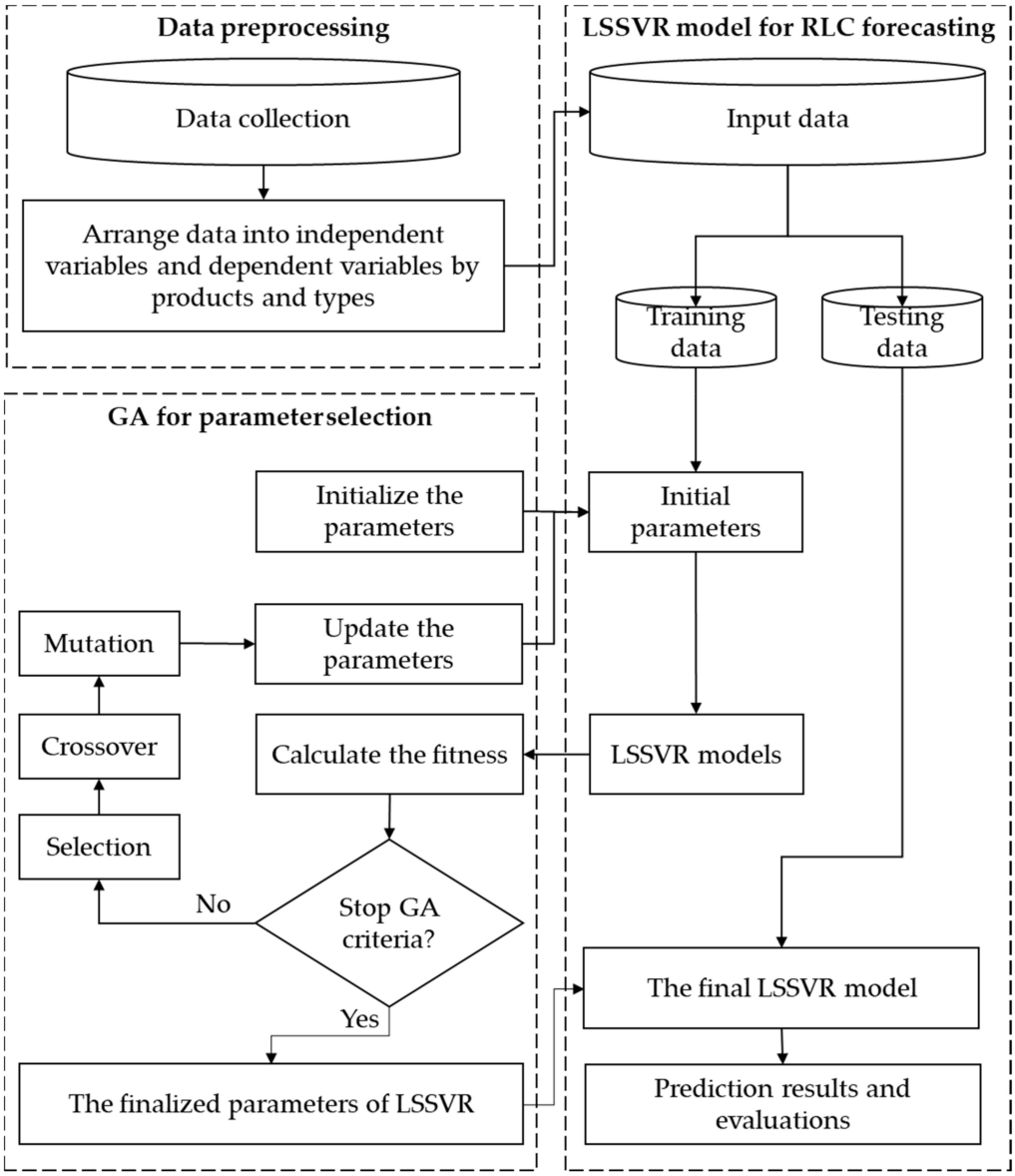


Figure 4. The framework of LSSVR-GA model for RLC values prediction.

Table 2. The features and samples in predicting RLC for different substrate products.

Data Sets	Features (X Variables)											Samples
	Ball	Bump	Base	L1	L2	L3	L4	L5	Via	Wire	Total	
FC_2L_T1	1	1	2	2					2		8 X	2232
FC_4L_T1	1	1	2	2	2	2			6		16 X	999
FC_6L_T1	1	1	2	2	2	2	2	2	10		24 X	742
WB_2L_T1	1		2	2					2	1	8 X	1400
WB_4L_T1	1		2	2	2	2			6	1	16 X	2704
WB_6L_T1	1		2	2	2	2	2	2	10	1	24 X	450

The preprocessed data sets were divided into 80% training data and 20% testing data. The training data were used to build the LSSVR model with the parameters optimized by GA. Before applying GA, it is necessary to encode the parameters to be optimized into a group of chromosomes. The common encoding methods include binary, real, multi-

objective, parallel, chaotic, and hybrid GA [30]. Considering the simplicity of implementation for factory operators, this study used the binary-coded GA to optimize the parameters of LSSVR, and each digital bit represented a gene. The length of the chromosome was defined according to the spatial range of the actual problem to be solved. The real number represented by the binary encoded was calculated as Equation (8).

$$RV = LB + \frac{\left(\sum_{i=1}^{l} d_i \cdot 2^{i-1}\right)}{2^l - 1} \cdot (UB - LB) \tag{8}$$

where RV is the real number represented by the binary encoded, LB is the lower bound of the spatial range, UB is the upper bound of the spatial range, l is the encoded bit length, and d_i is the bit value of the i^{th} bit.

Figure 5 shows the LSSVR model's encoded parameters—γ and σ—and the operation of real numbers. Each parameter consists of 10 genes, and the LSSVR model has two parameters. These two parameters represent 20 genes as a chromosome. The lower and upper bounds of the two parameters are both 1 and 500, and the real numbers represented are calculated accordingly. It is also necessary to define the optimized procedure settings of LSSVR-GA. The population size, iteration, crossover rate, and mutation rate were arranged at 40, 20, 0.8, and 0.1, respectively. When starting GA, the parameters must first be initialized as the input parameter of the LSSVR model. The training result of the LSSVR model is calculated by the fitness function. This is to evaluate the stopping conditions for GA. If conditions are not met, it will go through the process as in Figures 6–8, and the GA selection-crossover-mutation process will have a new generation. Good chromosomes have more opportunities to be selected. Unfit and less fit chromosomes are gradually eliminated. Therefore, the updated parameters are used as input parameters of the LSSVR model. To find the best-fit parameters of LSSVR, repeat the fitness function to compute the evaluation until the GA stop condition. Then, set the best-fit parameters of LSSVR in the final LSSVR model and perform RLC predictions.

Binary encoding	Hyperparameter1: γ (LB=1; UB=500)										Hyperparameter 2: σ (LB=1; UB=500)									
Chromosome	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20
	0	1	0	0	1	1	0	1	0	1	0	0	1	0	0	0	0	1	0	0
Operations that represent real numbers:	$1 + \frac{0 + 2^1 + 0 + 0 + 2^4 + 2^5 + 0 + 2^7 + 0 + 2^9}{2^{10} - 1}(500-1)$ = 337.5689										$1 + \frac{0 + 0 + 2^2 + 0 + 0 + 0 + 0 + 2^7 + 0 + 0}{2^{10} - 1}(500-1)$ = 65.3871									

Figure 5. The LSSVR parameter encoding and operation of representing real numbers.

Gene#	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20
Chromosome A	0	1	0	0	0	0	0	1	1	1	0	0	1	1	0	1	0	1	1	0
Represented parameters:	γ= 439.0274										σ= 209.7703									
Chromosome B	1	0	1	1	0	1	1	0	0	1	0	0	0	1	1	0	1	1	0	0
Represented parameters:	γ= 303.9120										σ= 106.3607									

Figure 6. The binary encoding for parameters of LSSVR at the stage of initial population.

Gene#	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20
Chromosome C	0	1	0	0	0	1	1	0	0	1	0	0	1	1	0	0	1	1	0	0
Represented parameters:	γ= 298.5464										σ= 100.5073									
Chromosome D	1	0	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	1	1	0
Represented parameters:	γ= 444.3930										σ= 215.6237									

Figure 7. The binary encoding for parameters of LSSVR after multi-point crossover.

Gene#	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18	G19	G20
Chromosome C	0	1	0	0	0	1	1	0	0	1	0	0	1	1	0	0	1	1	0	0
Represented parameters:	γ= 298.5464										σ= 100.5073									
chromosome E	0	1	1	1	1	1	1	0	1	0	1	1	1	1	0	1	0	0	0	0
Represented parameters:	γ= 187.3324										σ= 23.9257									

Figure 8. The binary encoding for parameters of LSSVR after mutation from chromosome C to E.

4. Numerical Results

Predicted results are evaluated and analyzed with the testing data to examine the effectiveness and interpretability of the proposed method. The evaluation is measured by mean absolute percentage error (MAPE (%)), weighted absolute percent error measure (WAPE (%)), and normalized mean absolute error (NMAE), as shown in Equations (9)–(11).

$$MAPE(\%) = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{\hat{Y}_i - Y_i}{Y_i}\right| \tag{9}$$

$$WAPE(\%) = 100\frac{\sum_{i=1}^{n}\left|\hat{Y}_i - Y_i\right|}{\sum_{i=1}^{n} Y_i} \tag{10}$$

$$NMAE = \frac{1}{max\ Y - minY}\left(\frac{1}{n}\sum_{i=1}^{n}\left|\hat{Y}_i - Y_i\right|\right) \tag{11}$$

where $\hat{Y}_i$ is the i^{th} predict value, Y_i is the i^{th} actual value, and i = 1~n.

Three other forecasting models with genetic algorithms, namely backpropagation neural networks (BPNN-GA), random forest (RF-GA), and eXtreme gradient boosting (XGBoost-GA), were employed to deal with the same data. Table 3 illustrates parameters determined by genetic algorithms to predict LCR values of different forecasting models. Lewis [31] reported that forecasting performance measured by MAPE values could be depicted in Table 4. Table 5 lists the MAPE, WAPE, and NMAE values of the four forecasting models.

The average performance of these six products is at the levels of good or highly accurate in predicting RLC using LSSVR-GA models. Furthermore, the LSSVR-GA models can generate average more accurate results than the other three forecasting models in terms of MAPE, WAPE, and NMAE.

Table 3. Parameters of forecasting models provided by genetic algorithms in LCR values prediction.

Method	Parameters	Ls (nH)	Cs (pF)	R (mΩ)	Ls (nH)	Cs (pF)	R (mΩ)	Ls (nH)	Cs (pF)	R (mΩ)
Products			**FC_2L_T1**			**FC_4L_T1**			**FC_6L_T1**	
GA-LSSVR	gamma	3.5241	220.2197	266.8764	386.9266	388.4704	336.2926	237.4034	386.9266	199.8153
	sigma	490.8269	2.3006	1.7176	1.4692	3.2476	4.0466	2.1650	1.4692	1.0638
GA-BPNN	learning rate	0.278163	0.725969	0.380188	0.316992	0.77501	0.668117	0.609052	0.168028	0.1674687
	momentum	0.790603	0.415267	0.424905	0.803841	0.748888	0.770531	0.864727	0.890341	0.186707
GA-RF	NTG *	162	122	313	281	268	423	491	316	159
	MTRY *	7	7	7	16	14	15	22	24	23
	NS *	32	25	13	9	11	9	10	5	11
	SSD *	3	5	6	7	15	12	23	16	23
	MN *	99	99	99	99	90	94	100	95	90
GA-XGBoost	CB *	0.9652	0.95019	0.94883	0.88056	0.85548	0.97795	0.83369	0.83964	0.95994
	SS *	0.73781	0.95313	0.9539	0.93509	0.93643	0.9374	0.9856	0.93392	0.9727
	MD *	9	10	9	9	9	9	10	9	7
	learning rate	0.0792	0.0981	0.08608	0.08813	0.08183	0.09519	0.04556	0.07721	0.07101
	gamma	0.03703	0.44082	0.9867	0.26736	0.00473	0.99912	0.0118	0.00621	0.19647
	MW *	4.67495	3.0224	3.60389	3.51156	3.60387	3.88513	3.15073	5.40802	3.43816
	Lambda *	0.84143	1.0274	0.73609	0.62597	1.06227	0.66675	0.58733	1.23425	0.66977
Products			**WB_2L_T1**			**WB_4L_T1**			**WB_6L_T1**	
GA-LSSVR	gamma	487.8397	461.7180	253.7229	260.8126	138.0979	291.6471	264.8891	386.9266	261.5888
	sigma	2.1064	1.6639	3.5726	2.1926	1.6762	2.3001	2.0693	1.4692	4.8404
GA-BPNN	learning rate	0.195829	0.558181	0.897148	0.270761	0.652679	0.405238	0.301158	0.86155	0.105814
	momentum	0.864213	0.65859	0.694771	0.894708	0.765861	0.88196	0.318845	0.639695	0.634643
GA-RF	NTG *	338	179	356	275	435	193	492	381	255
	MTRY *	7	7	7	14	11	13	22	24	21
	NS *	21	16	20	33	29	32	8	4	8
	SSD *	7	5	4	5	11	6	5	7	18
	MN *	97	100	99	99	100	98	99	96	85
GA-XGBoost	CB *	0.91655	0.88732	0.87073	0.98352	0.94273	0.82735	0.90711	0.81666	0.94269
	SS *	0.52606	0.70242	0.92529	0.66074	0.89115	0.91367	0.74513	0.90611	0.9466
	MD *	10	8	10	9	9	10	10	8	9
	learning rate	0.09689	0.06631	0.08815	0.09564	0.08211	0.08766	0.09008	0.09676	0.08981
	gamma	0.00929	0.02931	0.11065	0.01612	0.00016	0.1768	0.01459	0.01058	0.48457
	MW *	3.52074	5.63888	3.38181	4.18612	4.3746	3.62682	3.1284	3.44193	4.72415
	Lambda *	0.56806	1.17406	0.87922	0.78879	0.77885	0.52211	0.7232	1.13934	0.79814

* NTG: the number of trees to grow; MTRY: the number of variables used at each split; NS: the minimum size of terminal nodes; SSD: the sample sizes to draw; MN: the maximum number of terminal nodes trees in the forest can have; CB: subsample percentage of columns while generating new trees; SS: the subsample ration of training cases; MD: the maximum depth of the tree; MW: the minimum sum of weights related to child nodes; lambda: the L2 regularization term of weights.

Table 4. Levels of forecasting accuracy measured by MAPE [31].

MAPE Values (%)	Accuracy
<10	Highly accurate prediction
10–20	Good prediction
20–50	Reasonable prediction
>50	Inaccurate prediction

Table 5. Forecasting results of different models in terms of MAPE, WAPE, and NMAE.

Product Type	Method	Ls (nH)			Cs (pF)			R (mΩ)		
		MAPE (%)	WAPE (%)	NMAE	MAPE (%)	WAPE (%)	NMAE	MAPE (%)	WAPE (%)	NMAE
FC_2L_T1	GA-LSSVR	**16.88**	**15.63**	**0.04818**	**13.25**	**13.47**	**0.05634**	**6.20**	**6.48**	**0.02528**
	GA-BPNN	18.03	17.55	0.05410	13.93	13.49	0.05642	7.48	6.72	0.02623
	GA-RF	18.64	18.79	0.05792	13.43	13.44	0.05619	7.25	6.96	0.02712
	GA-XGBoost	19.01	19.49	0.06008	14.12	14.02	0.05862	8.74	7.77	0.03029
FC_4L_T1	GA-LSSVR	**12.18**	**8.92**	**0.01625**	**6.75**	**5.11**	**0.01527**	**12.15**	**5.99**	**0.00845**
	GA-BPNN	35.86	22.67	0.04130	9.81	8.12	0.02423	25.05	14.35	0.02024
	GA-RF	12.30	9.61	0.01751	8.60	6.95	0.02074	16.68	7.80	0.01100
	GA-XGBoost	15.16	11.06	0.02015	7.54	6.01	0.01796	17.85	7.87	0.01110
FC_6L_T1	GA-LSSVR	**10.35**	**7.81**	**0.05383**	**9.09**	**7.42**	**0.04804**	**11.99**	**8.33**	**0.04320**
	GA-BPNN	10.37	8.74	0.06019	9.26	8.63	0.05588	15.77	12.87	0.06674
	GA-RF	10.40	8.56	0.05897	9.46	8.35	0.05402	12.10	9.74	0.05053
	GA-XGBoost	10.87	9.15	0.06302	9.32	8.27	0.05352	12.23	10.45	0.05420
WB_2L_T1	GA-LSSVR	**13.28**	**12.63**	**0.05518**	**5.61**	**5.25**	**0.02018**	**7.21**	**6.79**	**0.02781**
	GA-BPNN	15.23	13.39	0.05852	6.64	6.00	0.02307	11.40	9.71	0.03980
	GA-RF	13.86	12.40	0.05417	6.10	5.86	0.02255	9.70	8.98	0.03680
	GA-XGBoost	13.42	11.48	0.05015	6.62	6.07	0.02337	9.05	8.05	0.03298
WB_4L_T1	GA-LSSVR	**14.54**	**11.65**	**0.03301**	**6.71**	**5.84**	**0.02074**	**10.00**	**7.10**	**0.02697**
	GA-BPNN	16.00	12.19	0.03453	9.19	7.53	0.02676	13.19	9.20	0.03497
	GA-RF	14.96	12.28	0.03480	9.16	7.79	0.02767	10.59	8.08	0.03071
	GA-XGBoost	15.81	11.74	0.03325	7.33	6.35	0.02255	10.10	7.47	0.02840
WB_6L_T1	GA-LSSVR	**8.68**	**9.28**	**0.05880**	**6.48**	**7.31**	**0.03904**	**4.08**	**4.89**	**0.03481**
	GA-BPNN	9.95	9.21	0.05838	6.56	6.63	0.03541	4.36	4.63	0.03299
	GA-RF	10.13	9.92	0.06289	6.61	6.79	0.03624	7.39	8.18	0.05819
	GA-XGBoost	9.60	8.75	0.05546	6.60	6.73	0.03593	6.71	7.43	0.05285
Average	GA-LSSVR	**12.65**	**10.99**	**0.04421**	**7.98**	**7.40**	**0.03327**	**8.60**	**6.60**	**0.02775**
	GA-BPNN	17.57	13.96	0.05117	9.23	8.40	0.03696	12.87	9.58	0.03683
	GA-RF	13.38	11.93	0.04771	8.89	8.19	0.03624	10.62	8.29	0.03573
	GA-XGBoost	13.98	11.94	0.04702	8.59	7.91	0.03533	10.78	8.17	0.03497

5. Conclusions

This study outlines an efficient method for predicting RLC circuit simulation in the IC package process using LSSVR-GA hybrid models. This method can be used to predict the integrity of RLC circuits. The ability to accurately predict analog circuits is essential to the IC package industry due to the timesaving in substrate design and process optimization. The numerical results revealed that the designed LSSVR-GA method is a feasible, effective, and efficient alternative for forecasting RLC values. For future research, one potential direction is to employ deep learning approaches to cope with the same data sets used in this study to improve forecasting accuracy. The other possible direction is to apply the presented LSSVR-GA framework to more complex circuit cases to examine performance.

Author Contributions: Conceptualization, Y.-P.W. and P.-F.P.; methodology, J.-P.L. and Y.-L.L.; software, J.-P.L. and Y.-L.L.; validation, H.-C.L. and C.-Y.S.; formal analysis, J.-P.L. and Y.-L.L.; data curation, H.-C.L. and C.-Y.S.; writing—original draft preparation, J.-P.L. and Y.-L.L.; writing—review and editing, P.-F.P.; visualization, J.-P.L. and Y.-L.L.; supervision, Y.-P.W. and P.-F.P.; funding acquisition, Y.-P.W. and P.-F.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Siliconware Precision Industries Co., Ltd., grant number 110A052.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The raw data of this work were supported by Huei-Chi Yang of Siliconware Precision Industries Co., Ltd.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Swaminathan, M.; Torun, H.M.; Yu, H.; Hejase, J.A.; Becker, W.D. Demystifying machine learning for signal and power integrity problems in packaging. *IEEE Trans. Compon. Packag. Technol.* **2020**, *10*, 1276–1295. [CrossRef]
2. Schierholz, M.; Sánchez-Masís, A.; Carmona-Cruz, A.; Duan, X.; Roy, K.; Yang, C.; Rimolo-Donadio, R.; Schuster, C. Si/pi-database of pcb-based interconnects for machine learning applications. *IEEE Access* **2021**, *9*, 34423–34432. [CrossRef]
3. Abdolrasol, M.G.; Hussain, S.S.; Ustun, T.S.; Sarker, M.R.; Hannan, M.A.; Mohamed, R.; Ali, J.A.; Mekhilef, S.; Milad, A. Artificial neural networks based optimization techniques: A review. *Electronics* **2021**, *10*, 2689. [CrossRef]
4. Ren, H.; Kokai, G.F.; Turner, W.J.; Ku, T.-S. Paragraph: Layout parasitics and device parameter prediction using graph neural networks. In Proceedings of the 2020 57th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 20–24 July 2020; pp. 1–6.
5. Shook, B.; Bhansali, P.; Kashyap, C.; Amin, C.; Joshi, S. Mlparest: Machine learning based parasitic estimation for custom circuit design. In Proceedings of the 2020 57th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 20–24 July 2020; pp. 1–6.
6. Wu, H.; Chu, W. Machine learning assisted structural design optimization for flip chip packages. In Proceedings of the 2021 6th International Conference on Integrated Circuits and Microsystems (ICICM), Nanjing, China, 22–24 October 2021; pp. 132–136.
7. Hsiao, H.; Chiang, K. Ai-assisted reliability life prediction model for wafer-level packaging using the random forest method. *J. Mech.* **2021**, *37*, 28–36. [CrossRef]
8. Lee, M.S.; Baick, I.; Kim, M.; Kwon, S.H.; Yeo, M.S.; Rhee, H.; Lee, E. Chip to package interaction risk assessment of fcbga devices using fea simulation, meta-modeling and multi-objective genetic algorithm optimization technique. In Proceedings of the 2021 IEEE International Reliability Physics Symposium (IRPS), Monterey, CA, USA, 21–25 March 2021; pp. 1–6.
9. Acharya, P.V.; Lokanathan, M.; Ouroua, A.; Hebner, R.; Strank, S.; Bahadur, V. Machine learning-based predictions of benefits of high thermal conductivity encapsulation materials for power electronics packaging. *J. Electron. Packag.* **2021**, *143*, 041109. [CrossRef]
10. Durgam, S.; Bhosale, A.; Bhosale, V.; Deshpande, R.; Sutar, P.; Kamble, S. Effective computational approach for optimization of temperature on printed circuit board. *J. Inst. Eng. Ser. C* **2022**, *103*, 763–776. [CrossRef]
11. Jing, S.; Li, M.; Li, X.; Yin, P. Optimization of reflow soldering temperature curve based on genetic algorithm. *Energy Rep.* **2021**, *7*, 772–782. [CrossRef]
12. Cecchetti, R.; de Paulis, F.; Olivieri, C.; Orlandi, A.; Buecker, M. Effective pcb decoupling optimization by combining an iterative genetic algorithm and machine learning. *Electronics* **2020**, *9*, 1243. [CrossRef]
13. Sourav, S.; Roy, A.; Cao, Y.; Pandey, S. Machine learning framework for power delivery network modelling. In Proceedings of the 2020 IEEE International Symposium on Electromagnetic Compatibility & Signal/Power Integrity (EMCSI), Reno, NV, USA, 28 July–28 August 2020; pp. 10–15.
14. Mao, M.; Wang, W.; Lu, C.; Jia, F.; Long, X. Machine learning for board-level drop response of bga packaging structure. *Microelectron. Reliab.* **2022**, *134*, 114553. [CrossRef]
15. Jin, H.; Gu, Z.-M.; Tao, T.-M.; Li, E. Hierarchical attention-based machine learning model for radiation prediction of wb-bga package. *IEEE Trans. Electromagn. Compat.* **2021**, *63*, 1972–1980. [CrossRef]
16. Wang, D.; Deng, L.; Zhang, L.; Zhang, F.; Qi, X.; Jin, H. Inverse design for wb-bga package structure by deep learning. In Proceedings of the 2021 13th Global Symposium on Millimeter-Waves & Terahertz (GSMM), Nanjing, China, 23–25 May 2021; pp. 1–3.
17. Kuo, C.-H.; Hu, A.H.; Hung, L.H.; Yang, K.-T.; Wu, C.-H. Life cycle impact assessment of semiconductor packaging technologies with emphasis on ball grid array. *J. Clean. Prod.* **2020**, *276*, 124301. [CrossRef]
18. Su, D.; Zhao, D.; Zhang, L.; Yang, H.; Wang, C.; Jiang, W. Reliability assessment of flip chip interconnect electronic packaging under thermal shocks. In Proceedings of the 2020 21st International Conference on Electronic Packaging Technology (ICEPT), Guangzhou, China, 12–15 August 2020; pp. 1–4.
19. Wang, D.; Zhao, Y.; Yao, Q.; Cao, Y.; Lian, B.; Zhang, H. Study on electrical performance of fc and wb in ic ceramic package. In Proceedings of the 2016 3rd International Conference on Materials Engineering, Manufacturing Technology and Control, Taiyuan, China, 27–28 February 2016; pp. 1670–1675.
20. Venkataiah, C.; Prasad, K.S.; Prasad, T.J.C. Effect of interconnect parasitic variations on circuit performance parameters. In Proceedings of the 2016 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 21–22 October 2016; pp. 1–4.
21. Khan, R.A.; Muhaisin, M.M.; Roberts, G.W. Extracting rlc parasitics from a flexible electronic hybrid assembly using on-chip esd protection circuits. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2021**, *68*, 4025–4037. [CrossRef]
22. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
23. Suykens, J.A.; Vandewalle, J. Least squares support vector machine classifiers. *Neural Processing Lett.* **1999**, *9*, 293–300. [CrossRef]

24. Suykens, J.A.; De Brabanter, J.; Lukas, L.; Vandewalle, J. Weighted least squares support vector machines: Robustness and sparse approximation. *Neurocomputing* **2002**, *48*, 85–105. [CrossRef]
25. Lodhi, H.; Saunders, C.; Shawe-Taylor, J.; Cristianini, N.; Watkins, C. Text classification using string kernels. *J. Mach. Learn. Res.* **2002**, *2*, 419–444.
26. Vert, J.-P.; Tsuda, K.; Schölkopf, B. A primer on kernel methods. *Kernel Methods Comput. Biol.* **2004**, *47*, 35–70.
27. Goldberg, Y.; Elhadad, M. Splitsvm: Fast, space-efficient, non-heuristic, polynomial kernel computation for nlp applications. In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers, Columbus, OH, USA, 15–20 June 2008; pp. 237–240.
28. Holland, J.H. *Adaptation in Natural and Artificial Systems*; The University of Michigan Press: Ann Arbor, MI, USA, 1975.
29. Holland, J.H. Genetic algorithms. *Sci. Am.* **1992**, *267*, 66–73. [CrossRef]
30. Katoch, S.; Chauhan, S.S.; Kumar, V. A review on genetic algorithm: Past, present, and future. *Multimed. Tools* **2021**, *80*, 8091–8126. [CrossRef] [PubMed]
31. Lewis, C.D. *Industrial and Business forecasting METHODS: A Practical Guide to Exponential Smoothing and Curve Fitting*; Butterworth-Heinemann: Oxford, UK, 1982.