IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 20, 2018, accepted May 30, 2018, date of publication June 11, 2018, date of current version July 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2845911

A Novel Trust Evaluation Method for Logic Circuits in IoT Applications Based on the E-PTM Model

JIE XIAO[1], JIANHUI JIANG[2], XIAOXIN LI[1], YUJIAO HUANG[1], XUHUA YANG[1], ZHANHUI SHI[1], AND JUNGANG LOU[3,4]

[1]College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China
[2]School of Software Engineering, Tongji University, Shanghai 201804, China
[3]College of Information Science, Huzhou University, Huzhou 313000, China
[4]Department of Computer Science, The University of Texas at San Antonio, San Antonio, TX 78249, USA

Corresponding author: Jungang Lou (loujungang0210@hotmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61502422, Grant 61772199, Grant 61432017, Grant 61503338, and Grant 61773348, and in part by the Natural Science Foundation of Zhejiang Province under Grant LY18F020028, Grant LY18F020031, Grant LY18F030023, and Grant LY17F030016.

ABSTRACT The increase in the reliability requirements of integrated circuits applied in diverse smart sensing devices and the increase in the cost of test generation and fault simulation have expanded the need for new approaches to estimate signal reliability in logic circuits, which will help trust management of Internet of Things smart systems. This paper presents a novel method for reliability analysis in logic circuits with unreliable devices for application in trust-driven design. Based on the extended probabilistic transfer matrix model with binary-decimal coding allocation, by using the technologies of state-vector expansion and matrix reconstruction, the proposed method evaluates the quality of a reliability improvement for trust-driven design applications, while maintaining high computational accuracy, in early stages of circuit design. This efficiency is possible, because the proposed method is always computed in units of basic gates and the reliability can be output by an observable matrix with hybrid coding. Simulation results on benchmark circuits show that the proposed method is an accurate and fast method with less complexity and will contribute to the dynamic analysis of circuit reliability in circuit design.

INDEX TERMS Binary-decimal code, state-vector expansion, matrix reconstruction, extended probabilistic transfer matrix model, circuit reliability (trust).

I. INTRODUCTION

With the wide application of internet of things (IoT) in our daily life, the security problems including the data security and reliability of the system attracted more and more attention [1], [2]. Among these challenges, the trustworthiness for big data collected by the diverse smart sensing devices containing logic circuits catches the extensive attention. However, new advances in the fabrication of logic circuits and scaling down their size to a few nanometers have resulted in logic circuits that are susceptible to failure. On the one hand, process variability, soft error and neutron particles are great threats to the very-large-scale integrated (VLSI) circuit in newer manufacturing technology [3]; on the other hand, shrinking the transistor size as well as lowering the supply voltage results in a significant reduction of the noise margin, which makes the circuits more prone to dynamic errors [4]. Therefore, the reliability (trust) of digital circuits in sensing devices is a critical issue in the design of new devices [5]–[8], and directly affects data trust in IoT applications.

To achieve a certain reliability at an acceptable cost, signal-reliability analysis for circuits is crucial because it can be used for practical problems such as fault detection, intrusion, test evaluation, functional reliability evaluation and signal probability evaluation of logic circuits [9]. The reliability is also an important research topic in other IoT applications and cloud environment [10]–[15]. Accuracy, scalability, computational complexity, memory requirements and single or multiple error occurrences are the main issues considered in a signal-reliability analysis.

In recent years, several approaches have been reported in the literature for signal-reliability analysis of logic

circuits [16]–[18]. According to their computational principles, the approaches roughly fall into three categories: (1) the analysis methods based on field data [19], the main shortcomings of which are over-sampling and lag evaluation; (2) the fault-injection methods [20], [21], which can be applied to multi-abstraction level logic circuits for signal-reliability analysis, and the simulations of which are highly accurate but very time intensive (a typical example is the Monte Carlo method); and (3) the analytical models, which are commonly used for signal-reliability analysis at one level of abstraction (typical examples include the probabilistic gate model (PGM) , Bayesian Network (BN) method [22], [23], and probabilistic transfer-matrix (PTM) model [24]). Compared with the fault-injection method, the analytical models have lower time complexities while maintaining high accuracies, but either their computational complexities are still too high or there are still some accuracy losses. Although they have many disadvantages, the analytical models have received more attention from industrial and academic circles because they exhibit good scalability and adaptability [7]. Therefore, the analytical model is chosen as the research object of this paper. In addition, the secret and optimization algorithms need to be combined into this system [25]–[27].

The PGM model calculates the signal probability of the primary outputs of the circuit by an iterative method in units of basic gates, but it ignores the situation in which the failure of the logical gates results in the repair of the circuit, which could easily cause some accuracy loss for circuits with signal correlation, and its runtime increases exponentially with the number of re-convergent fan-outs [5]. The signal probability of an output is defined as the probability that the output obtains a specified value, which is logic 1 in general [16].

The PTM model is one of the most credible methods for signal-reliability analysis in logic circuits and is useful in measuring the impact of path-based cumulative effects, such as glitch attenuation and logic masking, on error propagation [5]. However, it remains computationally intensive for large circuits due to the exponential time-space complexity with the number of primary inputs. Although the PTM model has some disadvantages, it can provide accurate results, which is what is needed in this paper. Therefore, the PTM model is chosen for this study to construct an accurate and fast signal-reliability analysis method for logic circuits.

To reduce the computational complexity of the PTM model, [28] presented a block-based method that decomposed a circuit into multiple blocks, calculated the results of the blocks using an accurate approach such as the PTM model, and then obtained the overall result using the reliability block-diagram method [29]. The main drawbacks of the method were high complexity for the block selection and accuracy loss for the decomposition. Using the level-matrix propagation method, [30] reduced the computation time of the PTM model by approximately 10%. In [31], macro-gates were proposed to reduce the memory requirement, but the computational cost was too high for circuits with multi-level nested fan-outs. Reference [18] solved the problem using the logic-partition strategy based on the associative law for tensor products, but its execution time was still large for modules with very large fan-out. Using a binary-based method, [32] could easily apply the effects of re-convergent paths to the calculated result. Although the accuracy of the method was very high, its computational complexity increased with the number of primary inputs. Reference [7] proposed an iterative calculation method based on the PTM model with hybrid coding, in which binary coding was used to describe signal relations, while decimal coding was used to quantify signal probabilities. That scheme had low computational complexity and was capable of analyzing large circuits, but with some accuracy loss. In short, the above methods focused only on the estimation of circuit reliability, which led to the neglect of some design requirements, such as the effects of the different input vectors on the output reliabilities in the leads. To meet the design requirements, [8] proposed a new analytical method based on the signal probability matrix [9]. In the method, all the input patterns for the whole circuit were coded in each primary input matrix and propagated to the corresponding input lead(s) of the post-stage gate(s) from primary inputs to primary outputs, which maintained its computational accuracy, but numerous unnecessary and repetitive calculations were performed. Moreover, the signal sources for gates in the circuit were not easy to extract accurately, which was not conducive to guiding circuit design in its early stage.

To avoid the disadvantages of the PTM method, many approaches [7], [18], [31], [33] with low computational complexity or sufficient accuracy were analyzed. In this paper, a method called extended probabilistic transfer matrix (E-PTM) is proposed to calculate the signal reliability of logic circuits. The probabilistic transfer matrix, which is used to quantify the non-deterministic behaviors of logic gates using a probabilistic method, is responsible for the accuracy of this method, while the binary-decimal coding mechanism, which is used to describe the signal states by a specified mode, is responsible for its computational complexity. The state-vector expansion and the matrix reconstruction, which are the main operations in this E-PTM model, are used to reduce the scale of the gate PTM without losing accuracy.

This rest of the paper is organized as follows. The E-PTM model is introduced in Section II. The probabilistic model for logic gates is introduced in Section III. The calculation algorithm based on the E-PTM model is described in Section IV. Simulation results on the circuits are analyzed and discussed in Section V. Finally, the paper is concluded in Section VI.

II. THE DESIGN OF THE E-PTM MODEL

To ensure the evaluation accuracy, the output result in each lead should fully cover all the possible states of the primary input(s) accessing the output lead with at least one path and the non-deterministic behaviors of the logic gates. To reduce the computational complexity, it is necessary to compute the

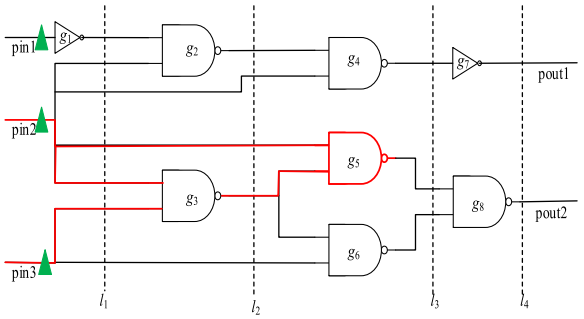


FIGURE 1. An example to illustrate the calculation characteristics of the E-PTM model.

output result in each lead in units of basic gates and avoid invalid calculations. To facilitate understanding, Figure 1 and formulas (1) and (2) are used to illustrate the calculation characteristics of the E-PTM model and the differences between the PTM models [24] and this model.

$$R_{circuit} = pin_{1\times2\times3} \times ((PTM_{l1} \otimes PTM_{l2} \otimes PTM_{l3} \otimes PTM_{l4}) \cdot \times (ITM_{l1} \otimes ITM_{l2} \otimes ITM_{l3} \otimes ITM_{l4})) \quad (1)$$

$$R_{g5} = f((PTM_{g5-1}^{pin_{2,3}} \otimes PTM_{g5-2}^{pin_{2,3}}) \times PTM_{g5}) \quad (2)$$

R_{gi} and $R_{circuit}$ are the output reliabilities of g_i and the circuit presented in Figure 1, respectively; $pin_{1\times2\times3}$ denotes the input probability distribution associated with the primary inputs of pin1, pin2 and pin3; PTM_{lj} and ITM_{lj} are the PTM and ITM (ideal transfer matrix) of the ljth layer of the circuit, respectively; PTM_{gi} is the PTM of gi; $PTM_{g5-1}^{pin_{2,3}}$ and $PTM_{g5-2}^{pin_{2,3}}$, which are associated with the primary inputs of pin2 and pin3, are the PTMs for the first and second inputs of g5, respectively; f is a spread function, which is presented in Section III.B; $1 \le i \le 8$ and $1 \le j \le 4$.

According to formulas (1) and (2), it can be known that the method presented in [24] was calculated in units of the whole circuit, which is inconvenient for calculating the output result in each lead and leads to large time-space complexity. The method presented in [16] was calculated in units of basic gates, but it contained all the irrelevant primary inputs in the calculations; for example, the irrelevant pin1 is included in the calculations of the output result of g_5, which leads to the method being suitable only for small circuits. The E-PTM model is calculated in units of basic gates and the calculations of the output results in the leads are only related to the primary inputs accessing the computing output with at least one path; an example is shown in Figure 1 and is marked with red lines.

To achieve the objectives, the proposed E-PTM model proposes a new signal matrix, which includes the input PTM and output PTM, to model the input-output relationship of the primary input signals and the traveling signals to ensure that the computation of signal reliability is performed in units of basic gates and excludes irrelevant operations. Moreover, the state-vector expansion and the matrix reconstruction are presented for the input PTM and the output PTM of basic gates so that the operations are performed between the correct elements to ensure the computational accuracy. The details of the implementation processes are presented in Section III.

III. PROBABILISTIC MODEL FOR LOGIC GATES

In this section, the calculation steps of the output PTM for a logic gate are presented. First, the primary input signals are initialized and their original sources are identified, and the non-deterministic behaviors of the basic gates in the circuit are quantified by using coding strategies. Second, the input PTM for the input leads of the basic gates is obtained by state-vector expansion and the input PTM of the corresponding basic gates is obtained by tensor product. Third, the output PTM of the basic gates is computed and reconstructed by a matrix product and some rules presented in [7], respectively. Finally, the iterative operations are performed from primary inputs to primary outputs in units of basic gates.

A. CODING STRATEGY

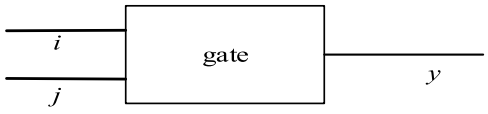
According to the above analysis, the input probability distribution (IPD) and the gate PTM in the logic circuits are the coding objects. The IPD is used to describe all possible states of the input signals for the logic gates in probability form, and the gate PTM is used to capture non-deterministic behaviors in the logic gates using a truth table.

To meet the calculation requirements of the proposed method, based on the framework presented in [7] and [16], a new input PTM with binary-decimal coding is proposed to simulate the IPD of logic gates. Its elements, in a matrix of size 2×2, are used to indicate the input signal probabilities corresponding to the four possible states: "correct 0", "error 1", "error 0" and "correct 1". On this basis, each element is extended to a vector to describe the contributions of all the input combinations to the corresponding output state, where the input(s) must be connected to the output with at least one path. In addition, the binary coding is appended to indicate the signal source(s) of the output in any leads in the circuit.

Further analysis found that the primary input signals are different from the traveling signals because the former are the driving sources of the logic circuit and must be initialized in advance. The input PTMs with hybrid coding are shown in Figure 2, and the input PTMs of the gates for the traveling signals, which are output from their pre-stage gates, are presented in Section III.B.

In Figure 2, a gate in a logic circuit with m primary input leads is taken as an example. pPM_i and pPM_j denote the input PTMs corresponding to the ith and jth primary input signals, respectively; each vector in the input PTMs is used to simulate the contribution of the corresponding primary input signal to the corresponding output state. ps_i and ps_j denote the fault probabilities corresponding to the ith and jth input signals, respectively; bci and bcj denote the binary codings corresponding to the ith and jth input signals of the gate, respectively, $bci = \underbrace{0}_{1} \cdots \underbrace{0}_{i-1} \underbrace{1}_{i} \underbrace{0}_{i+1} \cdots \underbrace{0}_{m}$,

$$\mathrm{pPM}_i = \begin{matrix} 0 \\ 1 \end{matrix}\begin{bmatrix} [1-ps_i,\ ps_i] & [ps_i,\ ps_i] \\ [ps_i,\ ps_i] & [ps_i,\ 1\text{-}ps_i] \end{bmatrix}_{bci}$$

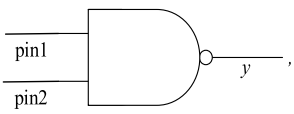


$$\mathrm{pPM}_j = \begin{matrix} 0 \\ 1 \end{matrix}\begin{bmatrix} [1-ps_j,\ ps_j] & [ps_j,\ ps_j] \\ [ps_j,\ ps_j] & [ps_j,\ 1\text{-}ps_j] \end{bmatrix}_{bcj}$$

FIGURE 2. An example for the input PTMs of the primary input signals.

$bcj = \underbrace{0}_{1} \cdots \underbrace{0}_{j-1} \underbrace{1}_{j} \underbrace{0}_{j+1} \cdots \underbrace{0}_{m}$, $1 \le i, j \le m$, and y is the output lead of the gate.

To capture non-deterministic behaviors in logic gates, the PTM framework based on the truth-table method is adopted. A NAND-2 gate with the inputs of pin1 and pin2 is taken as an example to introduce the creation of its gate PTM, as shown in Figure 3, where pg denotes the fault probability of the NAND-2 gate.



$$\mathrm{gPM} = \begin{matrix} 00 \\ 01 \\ 10 \\ 11 \end{matrix}\begin{bmatrix} pg, & 1-pg \\ pg, & 1-pg \\ pg, & 1-pg \\ 1-pg, & 1-pg \end{bmatrix}$$

FIGURE 3. The PTM of a NAND-2 gate.

B. THE INPUT PTM FOR GATES

The accuracy of the proposed method is mainly determined by the input PTM for gates and the gate PTM because logic gates are used as the basic units in the calculation process. The gate PTM can be accurately created by using the method introduced in Section III.A, so the remaining work is to present the accurate input PTM for gates.

For convenience and without loss of generality, taking a logic gate (g_k) with m_k input leads that is extracted from a logic circuit with n gates as an example, as shown in Figure 4, where aPM_k denotes its input PTM; ePM_{kh} denotes the input PTM corresponding to its hth input lead; ebc_{kh} and abc_k denote the binary codings corresponding to the ePM_{kh} and aPM_k, respectively; e_ele_{hd} and a_ele_d are the dth elements for ePM_{kh} and aPM_k, respectively; se_ele_{hdi1} coded by ebd_{hi1} and sa_ele_{di2} coded by abd_{i2} are the $i1$th sub-element for e_ele_{hd} and the $i2$th sub-element for a_ele_d, respectively; em_k and am_k are the numbers of '1' bits in ebc_{kh} and abc_k, respectively; oPM_k is the output PTM corresponding to its output lead; $1 \le k \le n$, $1 \le h \le m_k$, $1 \le d \le 4$, $1 \le i1 \le 2^{emk}$, $1 \le i2 \le 2^{amk}$; $e\_ele_{hd} = [\cdots \overset{ebd_{hi1}}{se\_ele_{hdi1}} \cdots]$, $a\_ele_d = [\cdots \overset{abd_{i2}}{sa\_ele_{di2}} \cdots]$.

It is known from reliability theory [18], [24] that the aPM_k can be obtained by using the tensor product of all the ePM_{kh}s, provided that the ePM_{kh}s have the same signal source(s); for example, for g_8 in Fig. 1, the signal sources of ePM_{81} and ePM_{82} are the same, namely, pin2 and pin3. However, there is a situation in which ePM_{kh1} and ePM_{kh2} of a gate often have different signal sources ($h1 \ne h2$, $h1, h2 \in [1, m_k]$). For example, for g_5, the signal sources of ePM_{51} and ePM_{52} are pin2 and (pin2, pin3), respectively.

To solve the problem of inconsistent signal sources, a method based on depth-first search [34] is proposed, and its main idea is as follows: first, extract the abc_k by performing bitor operations on all the ebc_{kh}s and identify the bit positions of ebc_{kh} in abc_k; then, expand all the ePM_{kh}s by using the depth-first search algorithm to make them have the same signal sources. The calculation process is presented in Algorithm 1.

Algorithm 1 An Expanded Method Based on Depth-First Search

Input: ePM_{kh} and the expanded-bit positions
Output: a new corresponding ePM_{kh}
1. Extract the corresponding ebc_{kh} from the ePM_{kh} for the gate g_k to obtain the corresponding abc_k of aPM_k using the bitor operation; obtain the array $\mathrm{pos}[em_k]$ for ebc_{kh} from abc_k using the bitand operation.
2. Extract ebd_{hi1} in ePM_{kh} and perform the following operations, $i1 = 1, 2, \ldots, 2^{emk}$.
   2.1. Extract abd_{i2} in aPM_k and perform the following operations, $i2 = 1, 2, \ldots, 2^{amk}$.
      2.1.1. Compare the $\mathrm{pos}[j]$ bit of abd_{i2} in aPM_k with the jth bit of ebd_{hi1} in ePM_{kh}, $j = 1, 2, \ldots, em_k$.
      2.1.2. If the comparative result is 1, then $sa\_ele_{di2} = se\_ele_{hdi1}$, $d = 1, 2, 3, 4$.
      2.1.3. $i2 = i2 + 1$.
   2.2. $i1 = i1 + 1$.
3. $\mathrm{ePM}_{kh} = \mathrm{aPM}_k$.

In Algorithm 1, step 1 obtains abc_k by one traversal operation of bitor among ebc_{kh}, and the array $\mathrm{pos}[em_k]$ by one traversal operation of bitand between abc_k and ebc_{kh}; ePM_{kh} and aPM_k need to be stored, while ePM_{k1} and aPM_k point to the same memory address in this step. Because abc_k and ebc_{kh} are both of length m, $h \in [1, m_k]$, the scale of ePM_{kh} and aPM_k is 2×2, and the scale of each element in ePM_{kh} and aPM_k is 1×2^{emk}. Therefore, the time complexity of step 1 is approximately $O((m_k + 1)^*m)$ and its space complexity is approximately $O(4^*m_k^*2^{emk})$. In step 2, an average of $(m\text{-}em_k + 1)/2$ comparisons are performed on each bit in ebd_{hi1}, while the length of ebd_{hi1} is em_k, $i2 \in [1, 2^{amk}]$ and $i1 \in [1, 2^{emk}]$, so the time complexity is approximately $O((m\text{-}em_k + 1)/2^*2^{amk+emk})$; in this step, only aPM_k needs to be stored, so its space complexity is approximately $O(4^*2^{amk})$. In step 3, only a pointer assignment is performed, so its time-space complexity is $O(1)$. To summarize the above

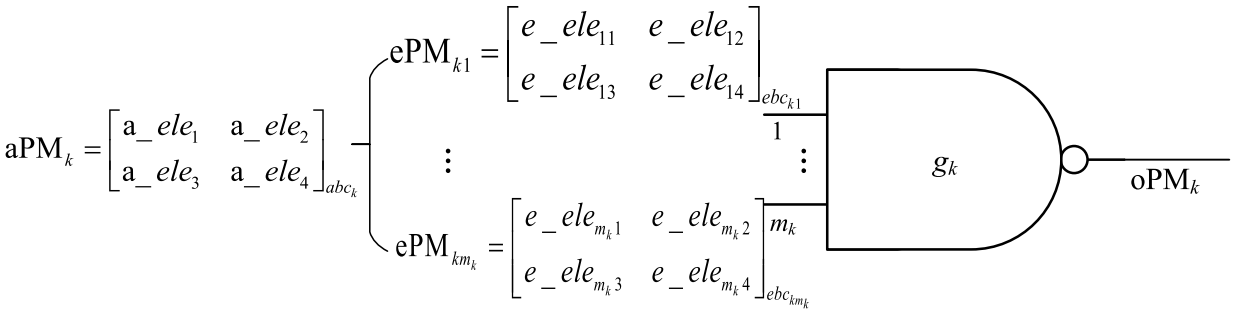


FIGURE 4. The relationship between ePM_{kh} and aPM_k.

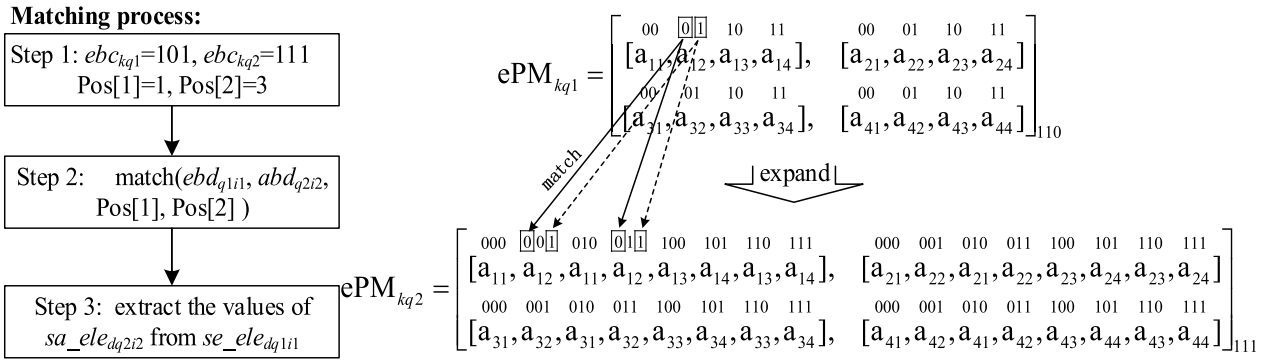


FIGURE 5. The matching process based on the depth-first search method.

analysis, the time complexity of the Algorithm 1 is approximately $O((m_k + 1)*m + (m - em_k + 1)*2^{amk+emk-1})$ and its space complexity is approximately $O(m_k*2^{emk+2} + 2^{amk+2})$. To further illustrate Algorithm 1, an example is presented in Figure 5.

Although ePM_{kh} can be precisely expanded, the time complexity of the depth-first-search-based method is too large, so a method called state-vector expansion is proposed and its details are as follows.

The main solution to the state-vector expansion is that the added signal source(s) should be fused into the current signal source(s) in ePM_{kh} in a specified order, which ensures that the operations are performed among the correct elements. For example, to resolve the inconsistency between ePM_{51} and ePM_{52}, pin3 should be added to ebc_{51} and each element in ePM_{51} needs to be recoded. The steps of the implementation strategy are given as follows; the expansion process and calculation flowchart of aPM_k are presented in Algorithm 2 and Figure 6, respectively.

First, identify the added signal source(s) for each input lead of the gates to expand its corresponding ebc_{kh}. To achieve this aim, we extract all the ebc_{kh}s of g_k and perform bitor operations on them to obtain abc_k of g_k.

Second, according to the abc_k of g_k obtained above, identify the expansion bit(s) of each ebc_{kh} and expand its corresponding ePM_{kh}. To reduce the computational complexity and ensure the expansion accuracy, a method based on shift operations is proposed.

Algorithm 2 State-Vector Expansion

Input: ePM_{kh} and the expanded-bit positions

Output: a new corresponding ePM_{kh}

1. Extract the corresponding ebc_{kh} from the ePM_{kh} for the gate g_k to obtain the corresponding abc_k of aPM_k by using the bitor operation; obtain the array pos[em_k] for ebc_{kh} from abc_k by using the bitand operation.
2. Obtain sa_ele_{di2} in aPM_k from ePM_{kh} by using shift operations, $i2 = 1, 2, \ldots, 2^{amk}$.
   - 2.1. initialize the identification $s = 0$;
   - 2.2. Perform $b = ((i2 - 1) \gg (am_k$ - pos[j])) & 1, then delete pos[j], $j = 1, 2, \ldots, em_k$;
   - 2.3. If $b = 1$, then $s = s \ll 1|\ 1$; else $s = s \ll 1$;
   - 2.4. $sa\_ele_{di2} = se\_ele_{ds}$, $d = 1, 2, 3, 4$;
   - 2.5. $i2 = i2 + 1$.
3. $ePM_{kh} = aPM_k$.

Third, obtain aPM_k through the tensor product of the ePM_{kh}s, which were obtained in the second step.

Here, sig is used to identify the expanded status of ePM_{kh} and aPM_k; $1 \le em_k \le am_k \le m$, $h \le t \le m_k$.

Similar to Algorithm 1, the time complexities of step 1 and step 3 for Algorithm 2 are approximately $O((m_k + 1)*m)$ and $O(1)$, respectively, their space complexities are approximately $O(4*m_k*2^{emk})$ and $O(1)$, respectively. In step 2,

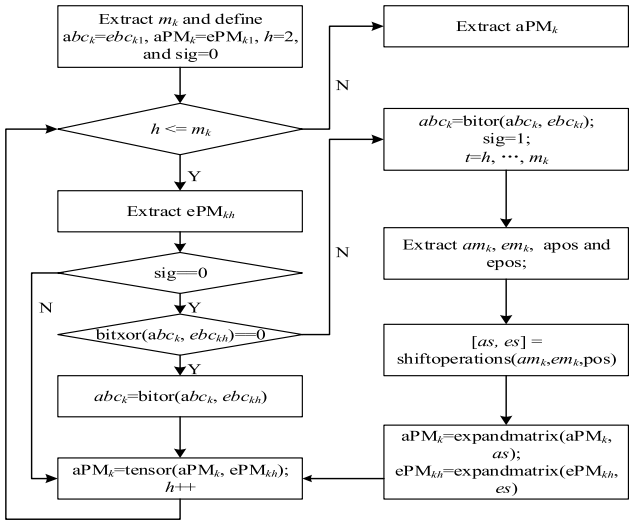


FIGURE 6. Calculation flowchart of aPM$_k$.

bitshift is a main operation to deduce the corresponding ebd_{ki1} through abd_{i2}, its average time complexity is approximately $O((am_k^2 + am_k)*2^{emk-1})$; in this step, only aPM$_k$ need to be stored, so its space complexity is approximately $O(4*2^{amk})$. To summarize the above analysis, the time complexity of the Algorithm 2 is approximately O($c_{k1}*2^{emk-1}$) and its space complexity is approximately O($m_k*2^{emk+2}+2^{amk+2}$), where c_{k1} is a parameter associated with am_k.

Obviously, the time complexity of Algorithm 2 is less than that of Algorithm 1, and they have the same space complexity.

From Figure 6, it can be seen that the main operations are tensor products, except for the state-vector expansion. The calculated object of the tensor product is g$_k$, which is a basic gate with m_k inputs and one output, so its time complexity is approximately $O(16*(m_k-1)*2^{amk})$ and its space complexity is approximately $O(2^{amk+mk+1})$. In state-vector expansion, the time complexity of the calculation of aPM$_k$ is approximately $O(c_{k1}*2^{emk-1}+c_{k2}*2^{amk})$ and its space complexity is approximately $O(m_k*2^{emk+2}+c_{k3}*2^{amk})$, where c_{k2} and c_{k3} are the parameters associated with m_k.

To further illustrate the state-vector expansion and the calculation of aPM$_k$, Figure 7 and Eq. (3) are shown as examples, respectively, where, ePM$_{kq1}$ and ePM$_{kr1}$ are the objects to be expanded; ePM$_{kq2}$ and ePM$_{kr2}$ are the expanded objects; $1 \le q1 = q2, r1 = r2 \le m_k$ and $q1 \ne r1$.

In Figure 7, to expand ePM$_{kq1}$ into ePM$_{kq2}$, the state-vector expansion is performed as follows: first, identify the positions of ebc_{kq1} in ebc_{kq2}; they are expressed as Pos[1]=1 and Pos[2]=2, which means that the first bit in ebc_{kq1} corresponds to the first bit in ebc_{kq2}, and the second bit in ebc_{kq1} corresponds to the second bit in ebc_{kq2}. Second, perform two shift operations on abd_{q2i2} in ePM$_{kq2}$ to obtain the corresponding ebd_{q1i1} in ePM$_{kq1}$; for example, '01' is obtained from the bitshift of '010'. Third, extract the sub-element in ePM$_{kq2}$ from ePM$_{kq1}$ by using ebd_{q1i1}; for example, a$_{12}$ coded by '010' in ePM$_{kq2}$ is extracted from a$_{12}$ coded by '01' in ePM$_{kq1}$.

The computational expression of aPM$_k$ is shown in Eq. (3), where ePM$_{kh}$ is the expanded result and $h = 1, 2, \ldots, m_k$.

$$\text{aPM}_k = \text{ePM}_{k1} \otimes \text{ePM}_{k2} \otimes \cdots \otimes \text{ePM}_{km_k} \tag{3}$$

The analysis indicates that Algorithm 2 is a simple procedure with high efficiency, for the following reasons: first, each sub-element in ePM$_{kh}$ or aPM$_k$ is regular, which is beneficial for extracting the final results by shift operations according to their identifications to avoid the time consumption of match operations presented in Algorithm 1. For example, to obtain the value coded by '010' in ePM$_{kq2}$, only two cycles of the shift operations need to be performed, as shown in Figure 7; second, for the elements in ePM$_{kh}$ and aPM$_k$, the same sequence is coded, so the shift operations only need to be performed on one element during the expansion of ePM$_{kh}$. For example, to expand ePM$_{kq1}$, we only need to perform the shift operations on one element to extract the values of all the elements in ePM$_{kq2}$.

C. THE OUTPUT PTM FOR GATES

The output PTM of a gate describes all the possible states of the output signal in probability form based on the truth table. It is known from the PTM model [24] that the output PTM of g_k (denoted as oPM$_k$) can be determined from its input PTM and its own gate PTM (denoted as gPM$_k$), and can be expressed by Eq. (4).

$$\text{oPM}_k = \text{aPM}_k \times \text{gPM}_k \tag{4}$$

The analysis found that the operation accurately reflects the possible states of the signal output from g_k. However, the order of the elements in oPM$_k$ is broken, mainly because logic gates are used as the basic units in the calculation process without accuracy loss. To meet the coding requirement of the input PTM belonged to its post-stage gate, we must reconstitute the coding of oPM$_k$, because the output PTM of g_k is equal to the input PTM of its post-stage gate. According to the coding rule of the input PTM of logic gates, which is presented in Section III.A, the oPM$_k$ is recoded by referencing the ITM of g_k, as shown in Algorithm 3.

Algorithm 3 Recoding oPM$_k$

Input: oPM$_k$ and the corresponding ITM
Output: oPM$_k$

1. Extract the sequence numbers with the states of correct 0, error 1, error 0 and correct 1 from the ITM of g_k, and put them into the queues Q$_{c0}$, Q$_{e1}$, Q$_{e0}$ and Q$_{c1}$, respectively.
2. Create a new output PTM and label it as oRPM$_k$.
3. Extract the corresponding elements from oPM$_k$ according to the sequence numbers in Q$_i$, perform the addition operation on them, and then put the result into the corresponding position in oRPM$_k$, i = c0, e1, e0, c1.
4. Delete oPM$_k$, Q$_{c0}$, Q$_{e1}$, Q$_{e0}$, Q$_{c1}$, and rename oRPM$_k$ as oPM$_k$.

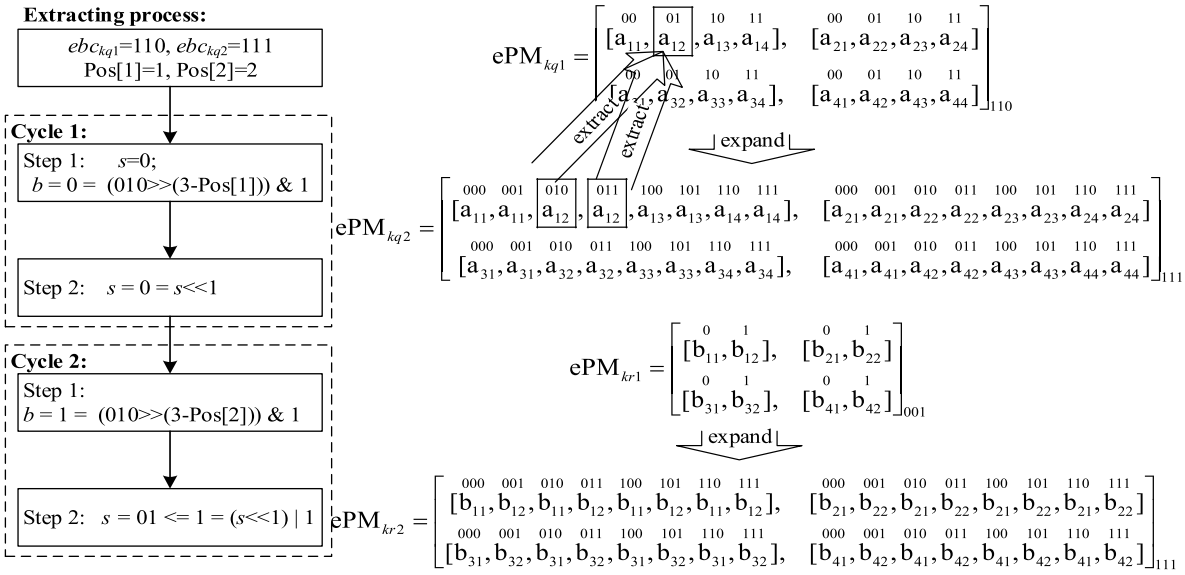


FIGURE 7. An example of state-vector expansion.

It is easy to know from Algorithm 3 that the time complexity for the reconstitution of oPM_k is approximately $O(2^{mk} + 2^{amk-1} + 4)$, and its space complexity is approximately $O(2^{mk+amk+1})$. Figure 8 is an example to show.

FIGURE 8. An example of the reconstitution of oPM_k.

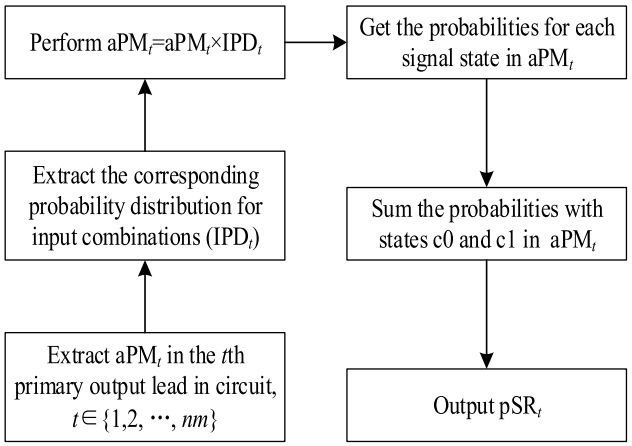


FIGURE 9. Calculation flowchart of the signal reliability in the primary output lead.

D. CIRCUIT SIGNAL RELIABILITY

The signal reliability of the logic circuit (denoted as cSR) is usually extracted at the primary output leads of the circuits, and its value is equal to the product of the signal reliabilities for the primary output leads [7], [31]. The signal reliability for the tth primary output lead (pSR_t) can be obtained by the steps shown in Figure 9, where mn denotes the number of primary output leads in the circuit and $1 \le t \le mn$.

IV. SIGNAL-RELIABILITY CALCULATION ALGORITHM

To achieve automatic computation, the signal reliability of the circuit is computed in two phases. In the first phase, the circuit is parsed to find m, n, and mn, and code the primary input signals as described in Section III.A; in the second phase, the gates in the circuit are extracted and processed one by one by using Algorithm 3, and then cSR is determined. The corresponding calculation flowchart for cSR is described in Figure 10 and the details of the calculation are presented in Algorithm 4.

In Algorithm 4, step 1 parses circuit netlist in units of basic gates, and each gate is stored to prepare for subsequent calculations, so its time-space is approximately $O(n)$. In step 2, its time complexity is approximately $O(n^*(c_{k1}{}^*2^{emk-1} + (c_{k2} + 1/2)2^{amk} + 2^{mk} + 4))$ on the basis of the time complexity presented in Section III; except for the essential information, only the input PTM, PTM and output PTM of the computing gate are stored during calculation, so the space complexity of step 2 is approximately $O(2^{mk+1} + m_k{}^*2^{emk+2} + 2^{amk+2}{}^*(2^{mk-1} + 1))$. For the primary outputs with mn, there is $mn - 1$ multiplications and only the final result should be stored, so the time complexity of step 3 is approximately

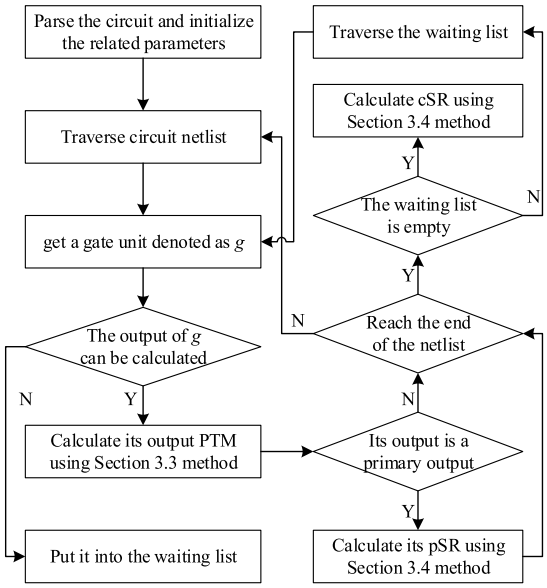


FIGURE 10. Calculation flowchart of cSR.

$O(mn-1)$ and its space complexity is approximately $O(1)$. Therefore, the time complexity of Algorithm 4 is approximately $O(n+n^*(c_{k1}{}^*2^{emk-1}+(c_{k2}+1/2)^*2^{amk}+2^{mk}+4)+mn-1)$, which can seem to be a linear increase with basic gates; its space complexity is about $O(n+2^{mk+1}+m_k{}^*2^{emk+2}+2^{amk+2*}(2^{mk-1}+1)+1)$, which can seem to be computed in units of basic gates for Algorithm 4.

Theoretical analysis shows that the accuracy of the proposed method is the same as that of the traditional PTM model presented in [24], while the time-space complexity of the proposed method is less than that of the traditional method, although some related parameters, such as input PTM and output PTM, are closely related to the primary inputs accessing the gate with at least one path, which could lead to an increase in the computation time and memory consumption of the proposed method, especially when the number of related primary inputs is increased.

Figure 11 illustrates an example of an implication of the proposed Algorithm 4. For convenience and without loss of generality, all the gates are assumed to have the same fault probability $p \in [0, 0.01]$, using the proposed method and the traditional PTM model, the output reliabilities for the circuit and each primary output lead are obtained, and then an analytical comparison of the calculation results is performed, as shown in Figure 12. In addition, the results for each lead are calculated by the proposed method when $p = 0.05$ to illustrate its flexibility, as shown in Figure 11.

According to Figure 11 and Figure 12, the proposed method computes in units of basic gates, which is beneficial for obtaining the output reliabilities of any leads in the circuit. Furthermore, the results obtained from this method are the same as the results obtained from the traditional PTM model for the circuit shown in Figure 11. These results indicate that the proposed method has the same calculation precision as the

Algorithm 4 Calculating cSR

Input: Circuit netlist
Output: cSR

1. Parse circuit netlist and initialize the related parameters.
   - 1.1. Extract the numbers of primary input leads and primary output leads, denote them as m and mn, respectively;
   - 1.2. Construct the encoding functions for the primary input signals and the gates in the circuit by using the method presented in Section III.A.
2. Calculate the output PTM of the circuit gates.
   - 2.1. Traverse the circuit netlist and get a gate unit;
   - 2.2. Get the input PTMs from the gate input leads. In case of failure, put the gate into the waiting list; else, calculate its input PTM by using the state-vector expansion method presented in Section III.B, and then obtain and recode its output PTM by the method presented in Section III.C;
   - 2.3. If its output lead is a primary output lead, calculate its output signal reliability by using the method presented in Section III.D;
   - 2.4. If reach the end of the netlist, go to step 2.5, else go to step 2.1;
   - 2.5. Traverse the waiting list and get a gate unit; if the waiting list is empty, go to step 3, else go to step 2.2.
3. Calculate the signal reliability of the logic circuit.
   - 3.1. If $mn = 1$, directly output the calculated result from step 2.3;
   - 3.2. If $mn > 1$, output the product of the calculated results from step 2.3.

traditional PTM model in the circuit shown in Figure 11. The reason dues to that the results obtained by the two methods include all possible states of the primary input signals, except that this method refers to the primary inputs accessing to the corresponding lead with at least one path, the traditional PTM model contains all the primary inputs of the circuit.

Compared with the traditional PTM model presented in [24], the main advantages of the proposed method are as follows: First, regarding the computational complexity, the proposed method computes in units of basic gates, while the traditional PTM model computes in units of the whole circuit. Second, the proposed method can be used to estimate the reliability of circuits with larger scales, thus enlarging the range of applicable circuit scales; examples are presented in Section V. Third, the proposed method expands the application scope, since it can be used to calculate the reliabilities of any leads in the circuits, which is favorable for modularization of circuit design. Moreover, the results of the proposed method can be output in multiple forms to meet the circuit design requirements; for example, according to the binary

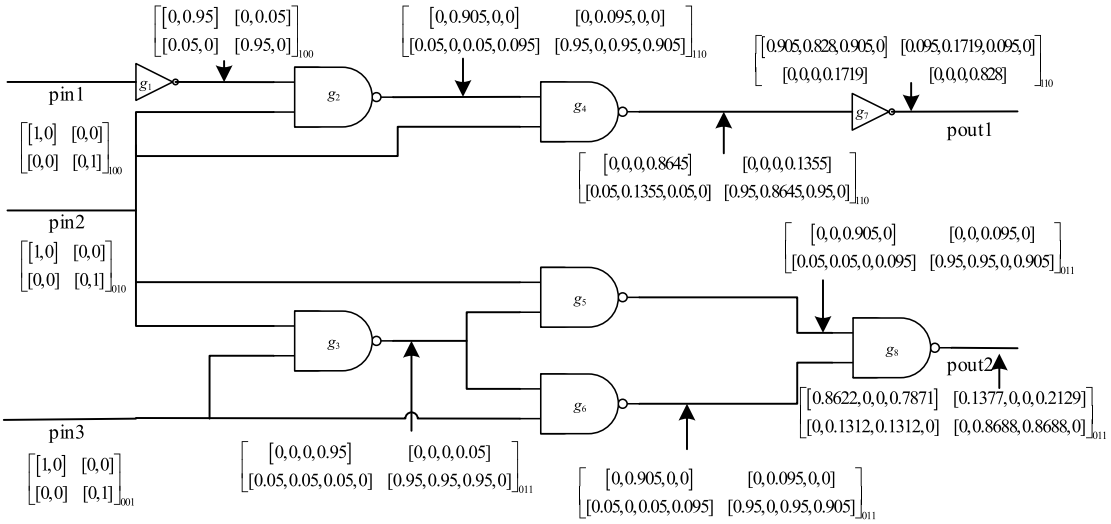


FIGURE 11. Example for a gate-level circuit.

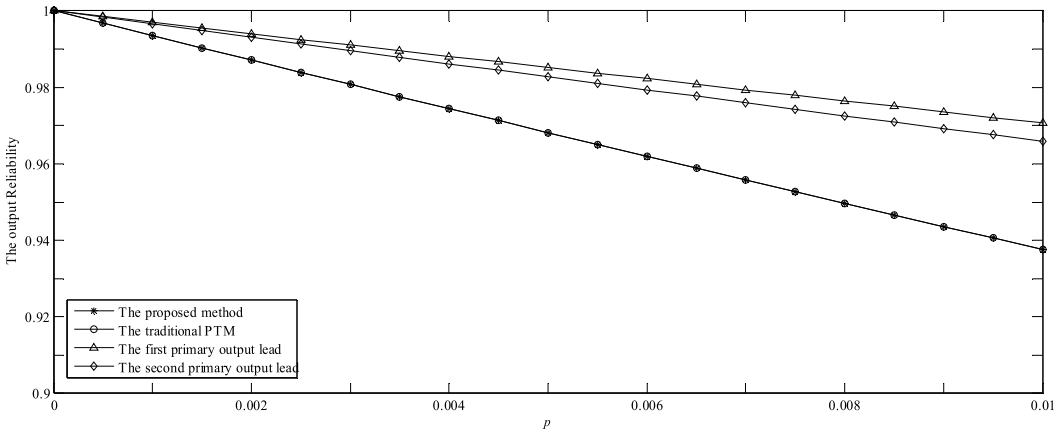


FIGURE 12. Comparison of the calculation results obtained by different methods.

coding of '011', we can easily determine that the results of pout2 in Figure 11 are independent of pin1.

V. SIMULATION EXPERIMENT

In this section, to demonstrate the accuracy and efficiency of the proposed method, simulations are performed for some typical circuits using a laptop computer with a 1.9 GHz processor and 4 GB RAM. For better comparison and analysis, the presented results are analyzed in three parts: the first part demonstrates the accuracy of this method, the second shows its efficiency compared with the Monte Carlo method presented in [35], and the last introduces some application examples.

In view of the accuracy of the PTM model and its acceptance in the fields, and the common practice of using the Monte Carlo (MC) method to verify the efficiency of the methods [7], [23], as well as the accuracy requirement of the proposed method, the PTM model and MC method are chosen to verify the efficiency of the proposed method in this paper, where the PTM model is applied to small-scale circuits, while the MC method is applied to 74-series benchmark circuits.

All of the primary input signals are in the ideal state and obey the uniform distribution, and all gates are assumed to have the same fault probability unless otherwise stated. The percentage relative error is calculated by using the following equation: Relative error = (Measured result − Reference value) / Reference value × 100%, where the reference value is obtained using the PTM model or Monte Carlo method.

A. ACCURACY

According to [7], [8], and [36], the traditional PTM model is an accurate reliability-analysis method and has higher precision than the Monte Carlo method, so its results are used as accurate reference values in this section. Considering that the PTM model applies only to small-scale circuits [36], for fair comparison, some small circuits, such as c17, full-adder and the circuit shown in Figure 11, are employed to compare the proposed method with the PTM model;

TABLE 1. Comparison of accuracy and computational complexity between the PTM model and the proposed method ($p = 0.001$).

Circuits	PTM model		The proposed method					
			DFS-based method			SVE-based method		
	Memory (MB)	Runtime (s)	Relative error (%)	Memory (MB)	Runtime (s)	Relative error (%)	Memory (MB)	Runtime (s)
C17	1.807	0.025	0	1.512	1.003	0	1.540	0.062
Full-adder	1.691	0.012	0	1.476	0.079	0	1.373	0.044
Schneider	1.711	0.061	0	1.301	0.515	0	1.332	0.170
Figure 11	1.795	0.014	0	1.464	0.087	0	1.328	0.046
Average results	1.751	0.028	0	1.438	0.421	0	1.393	0.081

TABLE 2. Comparison of accuracy and computational complexity between the Monte Carlo method and the proposed method ($p = 0.001$).

Circuits	Monte Carlo method		The proposed method					
			DFS-based method			SVE-Based method		
	Memory (MB)	Runtime (s)	Relative error (%)	Memory (MB)	Runtime (s)	Relative error (%)	Memory (MB)	Runtime (s)
74148	1.594	912.955	-0.480	2.687	49.072	-0.480	2.027	4.001
74155	1.243	127.590	-0.119	1.625	2.910	-0.119	1.710	0.483
74157	1.484	142.421	-0.224	1.589	3.138	-0.224	1.753	0.625
74181	2.385	6554.394	0.737	4.086	165.447	0.737	4.699	34.398
74182	1.270	526.9725	0.573	2.256	7.277	0.573	1.864	1.896
74185	1.709	2675.326	0.785	4.051	109.118	0.785	3.762	8.971
74283	1.801	1166.159	0.245	2.871	9.759	0.245	2.894	4.212
Average results	1.641	1729.403	0.217	2.738	49.532	0.217	2.673	7.798

the results are listed in Table 1, where the DFS-based method and SVE-based method refer to the proposed methods based on depth-first search and state-vector expansion, respectively.

According to Table 1, the comparison results show that the values provided by the proposed method are the same as the reference values obtained by the PTM model, and the results obtained by the DFS-based method and the SVE-based method are the same. The PTM model outperformed the proposed method in terms of time consumption, but its memory consumption was slightly higher on the employed circuits. The SVE-based method, on average, was 5.2 times faster than the DFS-based method, but their memory consumptions were similar. The reasons are as follows: first, all of the relevant input vectors were considered for each gate, and every possible behavioral characteristic of each gate was represented accurately in the methods. Second, to evaluate large-scale circuits, a new dimension with binary coding was added to the proposed method, which made it difficult to demonstrate the time advantage of the proposed method on the small-scale circuits. Third, the proposed method performed the calculation for reliability analysis in units of basic gates (as shown in Figure 11), while the traditional PTM model computed in units of the whole circuit. Moreover, the difference between the computational results obtained by the DFS-based method and the SVE-based method depended on the expanded difference in the input PTM; the corresponding analysis was presented in Section III.B.

B. EFFICIENCY

To demonstrate the efficiency of the proposed method, simulations were performed on 74-series benchmark circuits, including a four-bit carry-lookahead generator (74182) and a four-bit ALU (74181). Considering that the PTM model does not apply to large circuits and the Monte Carlo method is regarded as a reference method with reliable performance on different circuits with large numbers of simulations [7], [8], [18], to achieve effective verification, the Monte Carlo method was chosen to verify the efficiency of the proposed method in this section and 200,000 simulations were adopted for 74-series benchmark circuits. Table 2 presents the simulation results.

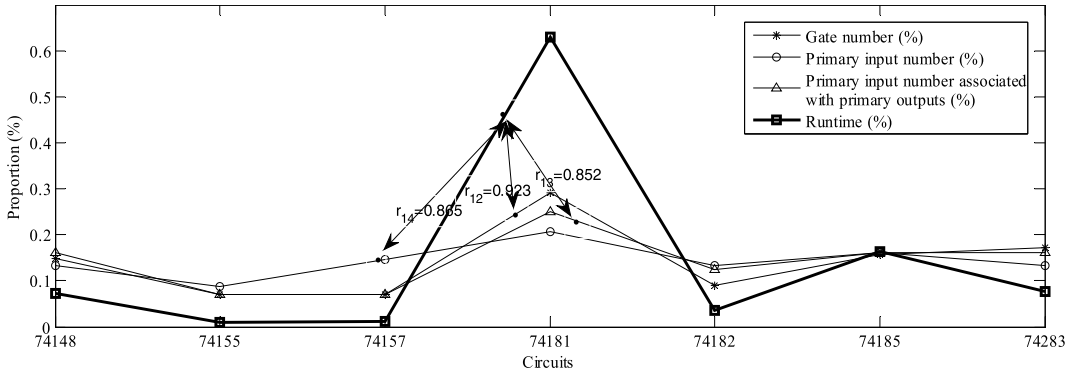


FIGURE 13. Comparison of computational times of the proposed method versus characteristics for several 74-series circuits.

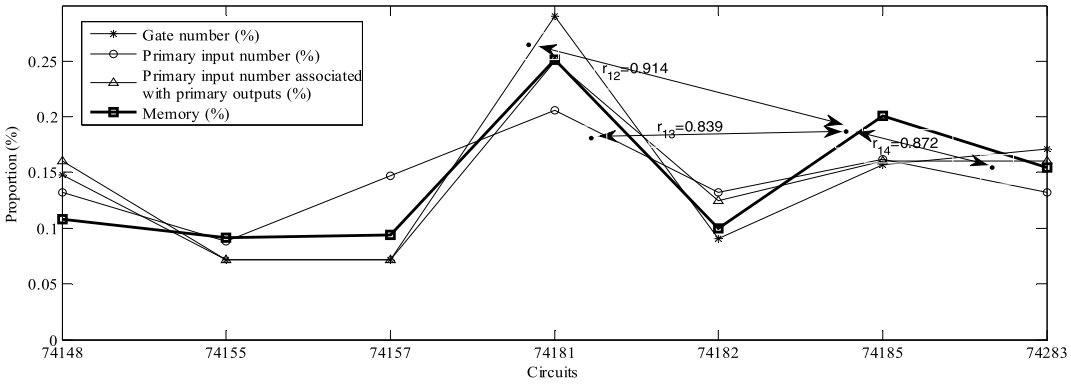


FIGURE 14. Comparison of memory utilizations of the proposed method versus characteristics for several 74-series circuits.

According to the relative errors presented in Table 2 and error theory in [37]–[39], it can be seen that the results obtained by the proposed method are very close to the results obtained by the reference approach, and the results obtained by the DFS-based method and the SVE-based method are the same. The computation time of the proposed method was far less than that of the reference approach. Compared with the memory consumption of the Monte Carlo method, the memory consumption of the proposed method had larger fluctuations and was greater. The SVE-based method, on average, was 6.35 times faster than the DFS-based method, but their memory consumptions were similar. The reasons are summarized as follows:

The proposed method is based on the PTM model, but its calculations are performed in units of basic gates and are only related to the primary inputs that access the computing gate with at least one path. The Monte Carlo method follows a pseudo-random strategy and always calculates in units of basic gates. Moreover, the Monte Carlo method mainly focuses on the computing gate during the calculations. For the large circuits, the SVE-based method outperformed the DFS-based method in terms of time consumption. Furthermore, circuits with more and larger fan-out branches tend to have larger time consumption, such as 74181 and 74185.

For further analysis, the required runtimes and the required memory of the proposed method on the 74-series benchmark circuits were plotted versus the number of circuit gates, the number of primary input leads and the maximum number of primary inputs associated with the primary output leads, as shown in Figure 13 and Figure 14, respectively. To facilitate presentation and comparison, these provided values were normalized and quantified by correlation coefficients to determine the major contributors.

Figure 13 and Figure 14 illustrate that the contributions to the computation time and the memory utilization of the proposed method, in order of importance, were the number of circuit gates, the maximum number of primary inputs associated with the primary output leads, and the number of primary input leads. These results indicate that the number of circuit gates can well reflect the complexity of the circuit structure, which was also reported in [40]. In addition to the number of circuit gates, the other two attributes mentioned above were also important factors, especially the maximum number of primary inputs associated with the primary outputs.

C. APPLICATION

Efficient reliability analysis, especially using accurate and fast methods that perform reliability analysis in units of

TABLE 3. Circuit reliability improvement using the reliability-critical gates ($p = 0.05$).

Method	Sorting results	Circuit-reliability increment	
This method	$g_7, g_4 > g_8, g_3 > g_6, g_5 > g_2, g_1$	$\Delta p_7=\Delta p_4=$ -0.01: ΔR=0.014	$\Delta p_8=\Delta p_3=$ -0.01: ΔR=0.0134
Ref. [28] method	$g_6, g_5 > g_3 > g_4, g_7 > g_8 > g_2 > g_1$	$\Delta p_6=\Delta p_5=$ -0.01: ΔR=0.0099	$\Delta p_3=\Delta p_4=$ -0.01: ΔR=0.0135
Ref. [7] method	$g_7, g_4 > g_8 > g_3 > g_6 > g_5 > g_2, g_1$	$\Delta p_7=\Delta p_4=$ -0.01: ΔR=0.014	$\Delta p_8=\Delta p_3=$ -0.01: ΔR=0.0133

circuit gates, can find many applications in reliability-driven circuit design. As an example, this section briefly describes how the proposed method can be used to verify the efficiency of some new methods, such as reliability analysis and importance measurement.

The proposed method has advantages in the following aspects: (1) computational accuracy, to avoid error identifying the reliability-critical gates to which the output reliability is very sensitive; (2) computational complexity, for strong adaptability to the large circuit modules; (3) the output reliability in each lead, to keep track of the output reliability in each lead during circuit design and facilitate timely decision making to lower costs; and (4) the primary inputs associated with the specified gate by at least one accessible path, which can be used to keep tabs on the contributions of each input vector to the output result to help the designers endow the test vectors with large fault coverage and consciously avoid some risks in practical application in the early stages of circuit design. The following is an application of this method, which was used to verify the effect of the approximate methods on the identification of the reliability-critical gates.

We took the circuit shown in Figure 11 again as an example. For a fair comparison, the approximate methods presented in [7] and [28], which have the same basic principle as the proposed method, were chosen to identify the reliability-critical gates, and the results are shown in Table 3, where it was assumed that all gates had the same fault probability p and the reliability-critical gates had the same increment (denoted as Δp); ΔR denotes the reliability increment of the circuit.

According to Table 3, the proposed method was better able to identify the reliability-critical gates in the circuit in Figure 11, compared with the approximate methods. The method presented in [7] performed better than the method presented in [28], while the results obtained by the proposed method were similar to those of the method presented in [7]. The main reason was that the method in [7] has much higher computational accuracy than the method in [28], so it was feasible to perform reliability analysis or importance measurement for circuits using the approximation method with higher precision, which could reduce the power consumption and speed up the calculation.

Moreover, using the proposed method, it was easy to identify the weak output leads (as shown in the circuit in Fig. 11) and accurately compute the sensitivity of each input vector to the primary outputs, where the sensitivity is the probability that the ideal and faulty outputs are different [4]. Take g_4 in the circuit in Fig. 11 as an example, through its output reliability distribution of $[0.95, 0.8645, 0.95, 0.8645]_{110}$, the following information could be extracted: (1) its signal sources were the primary inputs pin1 and pin2; (2) the sensitivities of the input vectors of 00, 01, 10 and 11 to the output of g_4 were 0.05, 0.1355, 0.05 and 0.1355, respectively; and (3) higher output reliability could be achieved when pin2=0 or the probability of pin2=0 was great. In summary, the above results depended on the computational characteristics of the method in this paper.

However, the analysis found that no current method could accurately calculate the reliability of a circuit with over 80 primary input leads with the existing computing power, even using supercomputers [41]. To achieve highly reliable circuit design at a small cost, modularity and approximate calculation are effective methods and can compensate for the disadvantages of the proposed method; this was also illustrated in Table 3. Further study of these issues is beyond the scope of this paper and shall be left as future work.

VI. CONCLUSION

The proposed method called E-PTM presented a state-vector expansion model with binary coding and shift operations to reduce the computational complexity of the PTM model and maintain its evaluation accuracy, which allowed us to accurately evaluate the reliability of large circuit modules in low computational time, exactly identify the output leads with weak reliability and easily obtain the sensitivities of the input vectors to each output lead. This made the proposed method very useful to the circuit designer and was helpful to verify the effectiveness of the related approximate calculation methods. Simulation results on benchmark circuits had shown the advantages of the proposed method in terms of accuracy and efficiency when compared with other methods, including the PTM model and Monte Carlo method. For example, the proposed method can be used to evaluate the reliability of some large circuits which are not applied for the PTM model, and it runs far faster than the MC method on some circuits, as shown in Table 2. The proposed method will play an important role in the early stages of circuit design according to its merits, which will help to promote device trust and data trust in IoT applications. In the future work,

the system's security will be considered, especially the use of homomorphic secrecy technologies [42]–[44].

REFERENCES

[1] X. Zhang, Y. Tan, C. Liang, Y. Li, and J. Li, "A covert channel over VoLTE via adjusting silence periods," *IEEE Access*, vol. 6, pp. 9292–9302, 2018, doi: 10.1109/ACCESS.2018.2802783.

[2] J. Shen, Z. Gui, S. Ji, J. Shen, H. Tan, and Y. Tang, "Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks," *J. Netw. Comput. Appl.*, vol. 105, pp. 117–123, Mar. 2018, doi: 10.1016/j.jnca.2018.01.003.

[3] Y. Du and S. Chen, "A novel layout-based single event transient injection approach to evaluate the soft error rate of large combinational circuits in complimentary metal-oxide-semiconductor bulk technology," *IEEE Trans. Rel.*, vol. 65, no. 1, pp. 248–255, Mar. 2016, doi: 10.1109/TR.2015.2427372.

[4] B. Srinivasu and K. Sridharan, "A transistor-level probabilistic approach for reliability analysis of arithmetic circuits with applications to emerging technologies," *IEEE Trans. Rel.*, vol. 66, no. 2, pp. 440–457, Jun. 2017.

[5] J. Xiao, J. Lou, J. Jiang, X. Li, X. Yang, and Y. Huang, "Blockchain architecture reliability-based measurement for circuit unit importance," *IEEE Access*, vol. 6, no. 4, pp. 15326–15334, Apr. 2018.

[6] V. H. Vaghef and A. Peiravi, "Node-to-node error sensitivity analysis using a graph based approach for VLSI logic circuits," *Microelectron. Rel.*, vol. 55, no. 1, pp. 264–271, Jan. 2015.

[7] J. Xiao, W. Lee, J. Jiang, and X. Yang, "Circuit reliability estimation based on an iterative PTM model with hybrid coding," *Microelectron. J.*, vol. 52, no. 4, pp. 117–123, Jun. 2016.

[8] C. Chen and R. Xiao, "A fast model for analysis and improvement of gate-level circuit reliability," *Integr., VLSI J.*, vol. 50, pp. 107–115, Jun. 2015.

[9] D. T. Franco, M. C. Vasconcelos, L. Naviner, and J.-F. Naviner, "Signal probability for reliability evaluation of logic circuits," *Microelectron. Rel.*, vol. 48, nos. 8–9, pp. 1586–1591, Aug. 2008.

[10] C. Yuan, X. Li, Q. M. J. Wu, J. Li, and X. Sun, "Fingerprint liveness detection from different fingerprint materials using convolutional neural network and principal component analysis," *Comput., Mater. Continua*, vol. 53, no. 3, pp. 357–371, 2017.

[11] Y. Liu, J. Ling, Z. Liu, J. Shen, and C. Gao, "Finger vein secure biometric template generation based on deep learning," *Soft Comput.*, vol. 22, no. 7, pp. 1–9, 2017.

[12] X. Chex, J. Li, X. Huang, J. Ma, and W. Lou, "New publicly verifiable databases with efficient updates," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 5, pp. 546–556, Sep./Oct. 2015.

[13] J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou, "Secure deduplication with efficient and reliable convergent key management," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 6, pp. 1615–1625, Jun. 2014.

[14] J. Li *et al.*, "Secure distributed deduplication systems with improved reliability," *IEEE Trans. Comput.*, vol. 64, no. 12, pp. 3569–3579, Dec. 2015.

[15] J. Shen, T. Zhou, X. Chen, J. Li, and W. Susilo, "Anonymous and traceable group data sharing in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 4, pp. 912–925, Apr. 2018.

[16] H. Zandevakili, A. Mahani, and M. Saneei, "An accurate and fast reliability analysis method for combinational circuits," *COMPEL, Int. J. Comput. Math. Elect. Electron. Eng.*, vol. 34, no. 3, pp. 979–995, May 2015.

[17] K. Lingasubramanian, "Probabilistic error analysis models for nano-domain VLSI circuits," Ph.D. dissertation, Dept. Elect. Eng., South Florida Univ., Tampa, FL, USA, 2010.

[18] J. Xiao, J. Jiang, and X. Zhu, "A method of circuit reliability estimation based on iterative PTM model," *Chin. J. Comput.*, vol. 37, no. 7, pp. 1508–1520, Jul. 2014.

[19] J. B. Bernstein, M. Gabbay, and O. Delly, "Reliability matrix solution to multiple mechanism prediction," *Microelectron. Rel.*, vol. 54, no. 12, pp. 2925–2951, Dec. 2014.

[20] K. Wu, H. Pahlevanzadeh, P. Liu, and Q. Yu, "A new fault injection method for evaluation of combining SEU and SET effects on circuit reliability," in *Proc. IEEE Int. Symp. Circuits Syst.*, Jun. 2014, pp. 602–605.

[21] D. Tang, C. He, Y. Li, H. Zang, C. Xiong, and J. Zhang, "Soft error reliability in advanced CMOS technologies-trends and challenges," *Sci. China Technol. Sci.*, vol. 57, no. 9, pp. 1846–1857, Sep. 2014.

[22] T. Li, J. Li, Z. Liu, P. Li, and C. Jia, "Differentially private naive Bayes learning over multiple data sources," *Inf. Sci.*, vol. 444, pp. 89–104, May 2018.

[23] T. Rejimon, K. Lingasubramanian, and S. Bhanja, "Probabilistic error modeling for nano-domain logic circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 17, no. 1, pp. 55–65, Jan. 2009, doi: 10.1109/TVLSI.2008.2003167.

[24] S. Krishnaswamy *et al.*, "Probabilistic transfer matrices in symbolic reliability analysis of logic circuits," *ACM Trans. Des. Automat. Electron. Syst.*, vol. 13, no. 1, p. 8, Jan. 2008, doi: 10.1145/1297666.1297674.

[25] Z. Liu, Y. Huang, J. Li, X. Cheng, and C. Shen, "DivORAM: Towards a practical oblivious ram with variable block size," *Inf. Sci.*, vol. 447, pp. 1–11, Jun. 2018.

[26] H. Wang, W. Wang, Z. Cui, X. Zhou, J. Zhao, and Y. Li, "A new dynamic firefly algorithm for demand estimation of water resources," *Inf. Sci.*, vol. 438, pp. 95–106, Apr. 2018.

[27] Y. Cao, Z. Zhou, X. Sun, and C. Gao, "Coverless information hiding based on the molecular structure images of material," *Comput., Mater., Continua*, vol. 54, no. 2, pp. 197–207, 2018.

[28] Z. Wang and J.-H. Jiang, "A serial method of circuit reliability calculation based on probabilistic transfer matrix," *Acta Electron. Sin.*, vol. 37, no. 2, pp. 241–247, Feb. 2009.

[29] O. Hasan, W. Ahmed, S. Tahar, and M. Hamdi, "Reliability block diagrams based analysis: A survey," in *Proc. AIP Conf.*, 2015, vol. 1648, no. 1, pp. 50129-1–50129-4.

[30] H. Ezzat and L. Naviner, "Level matrix propagation for reliability analysis of nano-scale circuits based on probabilistic transfer matrix," in *Proc. 11th Int. Symp. Quality Electron. Design*, 2010, pp. 524–527.

[31] J. Xiao, J. Jiang, X. Zhu, and C. Ouyang, "A method of gate-level circuit reliability estimation based on iterative PTM model," in *Proc. IEEE 17th Pacific Rim Int. Symp. Dependable Comput.*, Dec. 2011, pp. 276–277.

[32] H. Zandevakili, A. Mahani, and M. Saneei, "Reliability analysis of logic circuits using binary probabilistic transfer matrix," in *Proc. 21st Iranian Conf. Elect. Eng.*, 2013, pp. 1–6.

[33] J. Xiao, J. Jiang, and J. Liang, "Transistor-level oriented calculation of reliability for generalized gates based on PTM," *Sci. China Inf. Sci.*, vol. 44, no. 10, pp. 1226–1238, 2014.

[34] M. T. Goodrich, R. Tamassia, and D. M. Mount, *Data Structures and Algorithms in C++*. Hoboken, NJ, USA: Wiley, 2011, pp. 437–439.

[35] H. Janssen, "Monte-Carlo based uncertainty analysis: Sampling efficiency and sampling convergence," *Rel. Eng. Syst. Safety*, vol. 109, pp. 123–132, Jan. 2013.

[36] S. Krishnaswamy, I. L. Markov, and J. P. Hayes, "Design, analysis and test of logic circuits under uncertainty," Ph.D. dissertation, Dept. Comput. Sci. Eng., Michigan Univ., Ann Arbor, MI, USA, 2008.

[37] Z. Qian and G. Jia, *Error Theory and Data Processing*. Science Press, 2013.

[38] J. Lou, Y. Jiang, Q. Shen, and R. Wang, "Failure prediction by relevance vector regression with improved quantum-inspired gravitational search," *J. Netw. Comput. Appl.*, vol. 103, no. 2, pp. 171–177, Feb. 2018.

[39] J. Lou, Y. Jiang, Q. Shen, Z. Shen, Z. Wang, and R. Wang, "Software reliability prediction via relevance vector regression," *Neurocomputing*, vol. 186, pp. 66–73, Apr. 2016.

[40] J. G. Mcleish, "Enhancing MIL-HDBK-217 reliability predictions with physics of failure methods," in *Proc. Annu. Rel. Maintainability Symp.*, 2010, pp. 1–6.

[41] J. Pan. (2017). From the quantum entanglement to the quantum computer, mankind may untangle the origins of the universe at some point in the future. QQ Public Platform. [Online]. Available: http://www.360doc.com/content/17/0206/09/40145575_626842178.shtml

[42] Q. Lin, H. Yan, Z. Huang, W. Chen, J. Shen, and Y. Tang, "An ID-based linearly homomorphic signature scheme and its application in blockchain," *IEEE Access*, vol. 6, pp. 20632–20640, 2018, doi: 10.1109/ACCESS.2018.2809426.

[43] J. Xu, L. Wei, Y. Zhang, A. Wang, F. Zhou, and C.-Z. Gao, "Dynamic fully homomorphic encryption-based merkle tree for lightweight streaming authenticated data structures," *J. Netw. Comput. Appl.*, vol. 107, pp. 113–124, Apr. 2018.

[44] Q. Lin, J. Li, Z. Huang, W. Chen, and J. Shen, "A short linearly homomorphic proxy signature scheme," *IEEE Access*, vol. 6, pp. 12966–12972, 2018.

JIE XIAO received the Ph.D. degree in computer system architecture from Tongji University, Shanghai, China, in 2013. He is currently with the Department of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. He also serves as a Consultant and a Technical Adviser for a research institute in electronic information fields. His current research interests include reliability evaluation and fault-tolerant design, deep learning, and combinatorial optimization-computation.

JIANHUI JIANG received the B.E., M.E., and Ph.D. degrees in 1985, 1988, and 1999, respectively. He is currently a Full Professor of software engineering and the Vice Dean of the School of Software Engineering, Tongji University. He has co-authored two books and published over 180 technical papers. His current research interests include dependable systems and networks, software reliability engineering, VLSI/SoC testing, and fault-tolerance. He is a Senior Member of the Chinese Computer Federation (CCF). He is the Vice Director of the Technical Committee on Fault-Tolerant Computing, CCF. He has served on several program committees of national or international symposiums or workshops, including the IEEE Pacific Rim International Symposium on Dependable Computing, the IEEE Asian Test Symposium, and the IEEE Workshop on RTL and High Level Testing.

XIAOXIN LI received the Ph.D. degree in computer application technology from the South China University of Technology, Guangzhou, China, in 2009. Since 2009, he has been a Post-Doctoral Researcher with the Department of Mathematics, Faculty of Mathematics and Computing, Sun Yat-sen University, Guangzhou. He joined the Zhejiang University of Technology, Hangzhou, China, in 2013. His current research interests include deep learning, error coding, and image analysis.

YUJIAO HUANG received the B.S. degree in information and computer science, the M.S. degree in computational mathematics, and the Ph.D. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2008, 2010, and 2014, respectively. She is currently a Lecturer with the Zhejiang University of Technology. Her research interests are in areas of artificial neural networks, stability theory, and dynamical systems.

XUHUA YANG received the B.E. degree in automation from the China University of Petroleum, Dongying, China, in 1993, and the M.S. and Ph.D. degrees in control science and engineering from Zhejiang University, Hangzhou, China, in 2001 and 2004, respectively. He is currently a Professor of computer science and technology with the Zhejiang University of Technology, Hangzhou, China. His current research interests include artificial intelligence, complex network systems, intelligent transportation systems, link prediction, and deep learning.

ZHANHUI SHI is currently pursuing the master's degree with the Department of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. His current research interests include reliability evaluation and fault-tolerant design, deep learning, and combinatorial optimization-computation.

JUNGANG LOU received the B.S. degree in mathematics from Zhejiang Normal University, China, in 2003, and the M.Sc. degree in computational mathematics and the Ph.D. degree in computer science and technology from Tongji University, Shanghai, China, in 2006 and 2010, respectively. He was a Visiting Scholar with the Department of Computer Science, The University of Texas at San Antonio, from 2017 to 2018. He is currently an Associate Professor with the School of Information Engineering, Huzhou University, Huzhou, China. He also holds a post-doctoral position at the Institute of Cyber-Systems and Control, School of Control Science and Engineering, Zhejiang University, Zhejiang, China. His current research interests include dependable computing, reliability engineering, computer system performance evaluation, neural network optimization, and time series prediction.