Detection of IJTAG Attacks Using LDPC-based Feature Reduction and Machine Learning

Xuanle Ren[1,2], R. D. (Shawn) Blanton[1], and Vítor Grade Tavares[2]

[1]*Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA*
[2]*INESC TEC and Faculty of Engineering, University of Porto, Porto, Portugal*

***Abstract*—IEEE 1687 standard (IJTAG), as an extension to the IEEE 1149.1, facilitates efficient access to embedded instruments by supporting reconfigurable scan networks. Specifically, IJTAG allows each IP to be wrapped by a test data register (TDR) whose access is controlled by a segment insertion bit (SIB) or a scan-mux control bit (SCB). Because the TDRs and the SIB/SCB network are typically not public, but critical for accessing embedded instruments, they might be used for illegitimate purposes, such as dumping credential data and reverse engineering IP design. Machine learning has been proposed to detect such attacks, but the large number of instruments and parallel execution enabled by the IJTAG produce high-dimensional data, which poses a challenge to on-chip detection. In this paper, we propose to reduce the high-dimensional but sparse data using a low-density parity-check (LDPC) matrix. Experiments using a modified version of the OpenSPARC T2 to include IJTAG functionality demonstrate that the use of feature reduction eliminates 91% of the features, leading to 43% reduction in circuit size without affecting detection accuracy. Also, the on-chip detector adds moderate overhead ($\sim$ 8%) to the IJTAG.**

I. Introduction

The increasing complexity of integrated circuits (ICs) requires integration of a large number and variety of IPs. Learning the setup, integration and test procedure of each IP used to impose additional burden on IC designers and test engineers. To mitigate the resulting test issues, IEEE 1687 standard (IJTAG), as an extension to the IEEE 1149.1 (JTAG), has been developed to facilitate efficient access to embedded instruments [1]. IJTAG allows scan chains in each instrument to be configured using a segment insertion bit (SIB) or a scan-mux control bit (SCB). Specifically, a SIB gates an instrument, such that access to the instrument is possible only if the SIB is open; otherwise, the instrument is bypassed. An SCB controls a multiplexer that exclusively selects a scan chain. SIBs/SCBs may include a daisy chain or a hierarchical network that is accessed by operating the test access port (TAP) of the on-chip JTAG. The adoption of IJTAG not only reduces the effort of learning the setup, integration and test procedure, but also allows IPs to be modified or added locally without affecting other instruments.

This work was in part supported by the Project "NanoSTIMA: Macro-to-Nano Human Sensing: Towards Integrated Multimodal Health Monitoring and Analytics/NORTE-01-0145-FEDER-000016", which is financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF). This work was also supported by the Portuguese Foundation for Science and Technology, under Scholarship SFRH/BD/52166/2013, through the Carnegie Mellon Portugal Program.

Because of their accessibility via the JTAG port, IJTAG-compliant instruments may also be accessed illegitimately. For example, prior work has demonstrated that cryptographic keys can be derived by analyzing the data dumped from scan chains [2]. Specifically, cryptographic primitives use a block cipher to encrypt a plain text, but the intermediate results, usually located in scan chains, can be shifted out, through which the key can be derived. An attacker may also reverse engineer the SIB/SCB network and the meaning of each bit in TDRs, which can then be used to derive data from on-chip memory [3], update firmware [4], and control chip operation [5]. Because IJTAG has gained growing support from EDA vendors, and is projected to be adopted widely in industry [6], the security of IJTAG is therefore a topic of importance.

Various countermeasures have been proposed to protect the IJTAG, including access restriction, encryption, attack detection and obfuscation. Access to the IJTAG can be restricted through fusing off the JTAG port, or monitoring if a user attempts to assert the SIBs that gate secure scan chains [7]. Both methods disable access to entire or parts of a scan chain, and therefore hinder in-field debugging and programming. The work of [8] proposes to insert key bits into scan chains such that the SIBs are opened only if a correct key is supplied. The use of an LFSR [9] and honey-traps [10] makes the key more complex. However, the key might be leaked during distribution [11]. The work in [12] improves the security of the IJTAG using a challenge-response protocol, which, however, requires availability and security of network communication. Another countermeasure involves detection of illegitimate IJTAG access through checking if the number of shifting cycles exceeds a pre-defined range [13]. This simple, rigid rule, however, is not able to detect complex attacks, and may also result in many false positives [14], [15].

Although machine learning proves to be an effective approach [14], [15], detection of illegitimate IJTAG access is more challenging than the JTAG, primarily for two reasons. First, characterizing IJTAG operation may require many more features. Different from JTAG operation that can be characterized using the sequence of instruction-register opcodes [15], IJTAG operation involves configuring a hierarchy of SIBs/SCBs and setting/resetting the bits in a large number of TDRs. Second, the IJTAG not only allows simultaneous assertion of multiple SIBs/SCBs, but also allows simultaneous setting of multiple bits in a TDR. This implies that many more combinations of operations become possible compared

978-1-5386-3728-9/18/$31.00 ©2018 IEEE

to the JTAG. This paper proposes an approach to reduce the high-dimensional data collected from IJTAG operation through compressed sensing. Reduction is possible because IJATG operation is likely to result in a sparse data set. Specifically, a low-density parity-check (LDPC) matrix is chosen for reducing the sparse data because it exhibits a good trade-off between compressed sensing and hardware overhead. The detection method assumes that an attacker only has access to the JTAG port and is, at least initially, unaware of the IJTAG architecture. This work has three contributions:

- The OpenSPARC T2 is enhanced by including eleven IJTAG-compliant instruments that are taken from commercial IC designs [16].
- After reviewing the reported threats to the JTAG and IJTAG, attacks of the modified OpenSPARC T2 are constructed based on those reviews.
- The use of LDPC-based feature reduction eliminates 91% of the features, and reduces circuit size by 43% without affecting detection accuracy, according to experiments based on the modified OpenSPARC T2.

The rest of this paper is organized as follows. In Section II, the background of IJTAG and illegitimate JTAG accesses are reviewed. Section III elaborates upon the process of IJTAG attack detection. Section IV presents the modification of the OpenSPARC T2, and evaluates the performance of the feature reduction. Section V discusses several issues concerning the proposed approach, and Section VI concludes the paper.

II. IJTAG and IJTAG Attacks

Fig. 1 shows an example IJTAG architecture with four instruments. The access to an instrument is configured using SIBs/SCBs. A SIB or an SCB consists of two flip-flops (FFs), for shifting and updating. To access a target TDR, the user needs to operate the TAP controller to shift in a logic one to its gating SIB, and/or shift in a proper value to its selecting SCB. After the UPDATE-DR state loads the values into the updating FF of the SIBs/SCBs, the target TDR is configured as a part of the chain. For example, to access the TDR of MEM1 in Fig. 1, the user needs to operate the TAP controller and supply the proper opcode into the instruction register (IR) that selects the SIB/SCB network configuring MEM1. Now the TAP controller has access to a four-bit chain, i.e., SIB1→SCB2→SIB3→SIB4. Next, the user needs to shift in '1100' during the SHIFT-DR state (the LSB is shifted in first) and update the value to the SIBs/SCBs during the UPDATE-DR state. Now the TDR of MEM1 is configured within the scan chain, which means the TAP controller now can access it.

A TDR, consisting of a shift register, can be accessed via a standard eight-port interface, as shown in Fig. 2. A serial data stream is shifted into the instrument via the TDR_SIN port and shifted out via the TDR_SOUT port. The other ports are controlling signals decoded from the TAP controller. The TDR can supply test vectors or debugging commands to the instrument when the TDR_UPDATE is set, and capture test response data when the TDR_CAPTURE is set. Fig. 2 shows an industrial example of memory wrapped by a six-bit TDR

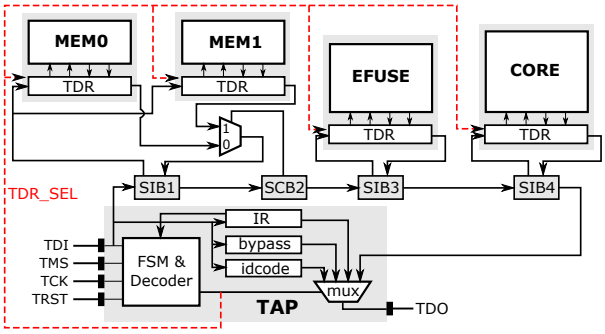


Fig. 1. An example IJTAG architecture composed of four instruments, each of which is gated by a SIB and/or selected by an SCB.

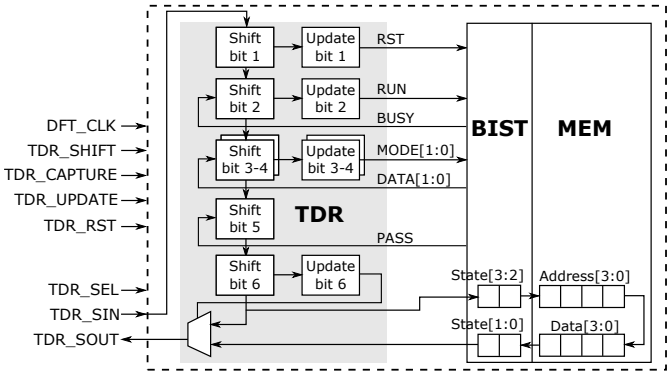


Fig. 2. An industrial memory macro wrapped by a six-bit TDR [16].

[16]. Bit one is used for resetting the BIST (built-in self-test). Bit two is used for starting the BIST and capturing the status of the BIST (busy or not). Bits three and four are used for updating a new BIST mode and capturing the existing BIST mode. Bit five only has a shift cell since it can only be used for capturing the BIST result (pass or not), but not for updating. Bit six is used like a SIB/SCB, because it can be used for accessing the next level of registers. Since these bits can be set simultaneously, their corresponding operations can also be executed in parallel. For example, a simultaneous setting of bits three to five indicates the parallel execution of selecting the BIST mode and querying the BIST result.

To access the instruments, an attacker would first need to reverse engineer the SIB/SCB network. This can be achieved by tentatively loading one and zero to each bit, and observing if the length of the chain between the TAP TDI and TDO changes. The attacker can repeat this interrogation until the whole SIB/SCB network is reverse engineered. Once gaining access to a TDR, the attacker can uncover the operation corresponding to each TDR bit, exploiting strategies similar to those described in [14]. More precisely, an attacker can set each TDR bit and check the response. The attacker can also vary the sequence of bit setting, and examine possible interactions between them. Further, the parallel execution enabled by the IJTAG makes functional reverse engineering more efficient, since the attacker can set multiple bits simultaneously, other than sequentially, for checking their interaction.

III. METHODOLOGY

This section describes the features used for characterizing IJTAG operation, reduction of the features, and attack detection using a random forest classifier.

A. Feature extraction

As described in Section II, IJTAG allows simultaneous setting of multiple bits in a TDR, which means IJTAG operations can be performed in parallel. This typically happens in two scenarios, namely independent operation and pipelining operation. TDR bits are dependent if swapping their order leads to a different result. For example, bit two and bits three to four in Fig. 2 are dependent because a different BIST mode (updated through operating bits three to four) may initialize a different BIST process (operated by setting bit two). Dependent bits are typically not set simultaneously unless they are operated in a pipelining manner. For example, bit two and bits three to four, if set simultaneously for a pipelining use, means that the BIST mode is set for the next BIST process rather than the current one. Besides dependent bits, TDR bits may even conflict if setting them simultaneously causes uncertain results. For example, bit one, used for resetting the BIST, conflicts with bit two which is used for starting the BIST. Although this conflict can be handled through a careful design, simultaneously initiating operation should be avoided.

IJTAG operation, consisting of SIBs/SCBs, TDRs and the TAP controller, is characterized using the features shown in Table I (N_S and N_T refer to the number of SIBs/SCBs and the number of TDRs, respectively). These features are collected during the UPDATE-DR state. To better reflect the sequential feature of IJTAG operation, a sequence of cycles, rather than a single cycle, are monitored (a cycle is defined as the period between two consecutive UPDATE-DR states). Specifically, data are collected using a sliding window (with w cycles) in an overlapping manner.

B. Compressed sensing

The data collected using a sliding window typically have a large dimension, but they are likely sparse. This is because the operations corresponding to the TDR bits, in most cases, should still be executed in specific orders. A simultaneous setting of all TDR bits (or most of them) is not practical. This observation means that the dimensionality of the data might be reduced. A technique of dimension reduction is fixed-length encoding (i.e., encode each possible combination of the observed data using a fixed-length code, whose length depends on the number of combinations in the observed data). However, fixed-length encoding may not reduce the dimension effectively because most combinations, although absent from the training set, are still possible. Another technique involves compressed sensing that aims to compress and reconstruct high-dimensional data with low complexity [17]. Let $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, and $A \in \mathbb{R}^{m \times n}$ ($m \ll n$), the compression and reconstruction of x can be formulated as

$$y = Ax \tag{1}$$

TABLE I
FEATURES USED FOR CHARACTERIZING IJTAG OPERATION.

Category	Index	Description	Count
SIBs/SCBs	F1	SIB/SCB bits	N_S
	F2	No. of asserted SIBs/SCBs	1
	F3	No. of bit transitions[1] in SIBs/SCBs	1
TDRs	F4	No. of ones in each TDR	N_T
	F5	No. of bit transitions in each TDR	N_T
	F6	No. of dependent bits in TDRs	1
TAP	F7	No. of TMS transitions	1
	F8	TEST-LOGIC-RESET activated?	1

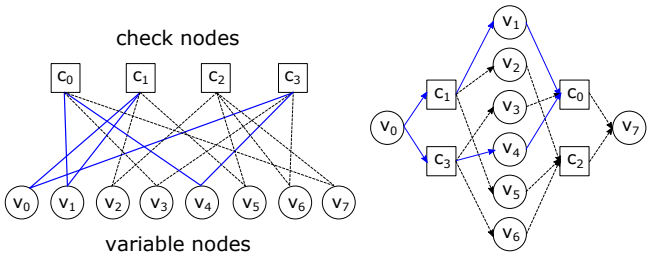
and

$$\hat{x}_{Lasso} := \underset{x}{\operatorname{argmax}} \|y - Ax\|_2^2 + \lambda \|x\|_1 \tag{2}$$

respectively. The compression is simply a multiplication of x and a matrix A representing a linear transformation from $\mathbb{R}^n$ to $\mathbb{R}^m$. The reconstruction involves a linear regression using Lasso regularization (i.e., $\|x\|_1$). Note that although reconstruction is not necessary for classification, it is still beneficial because it evaluates the quality of the compression. The linear regression formulated in (2) is underdetermined ($m \ll n$), meaning that the solution of $\hat{x}$ is not unique. To achieve a sparse solution of $\hat{x}$, the Lasso regularization, rather than $\|x\|_2^2$, is used. However, the performance of the Lasso declines if the columns of A are highly correlated (in this case, the Lasso simply chooses one column of A). To mitigate this problem, the matrix A should satisfy the restricted isometry property (RIP) that requires a matrix to be "almost" orthonormal, at least when operating upon sparse vectors. However, there exist no effective approaches to construct such matrices, although some matrices, like Gaussian matrices, satisfy the RIP with exponentially high probability [18].

As studied in [19], LDPC codes, originally used as error correcting codes, show an outstanding performance when used for compressed sensing. An LDPC code can be represented using a binary matrix (typically called the LDPC matrix) or a bipartite graph (which is also referred to as a Tanner graph) [20]. Fig. III-B shows an example of a Tanner graph that represents the same LDPC code as the matrix A in (3). The graph consists of n variable nodes (the number of bits in a code word) and m check nodes (the number of parity bits). Check node c_i is connected to variable node v_j if the element a_{ij} of A is a 1. The Tanner graph shown in Fig. III-B can also be represented using a tree that is constructed by traversing the adjacent nodes non-repeatedly as shown in Fig. III-B. Note that, starting from the root node (i.e., v_0), there exist many cyclic paths, among which the length of the shortest path is defined as local girth, g. A large value for g means that the root node is significantly independent from other variable nodes, and therefore is more likely to lead to an orthonormal matrix. Besides, because the entries of an LDPC matrix are either one or zero, feature reduction using an LDPC matrix involves solely additions, rather than matrix multiplication.

[1] A bit transition refers to a SIB/SCB changing from asserted to de-asserted, or from de-asserted to asserted.



(a) (b)

Fig. 3. (a) An example of a Tanner graph with eight variable nodes and four check nodes. (b) The adjacent variable nodes and check nodes of the Tanner graph in Fig. III-B are traversed starting from v_0.

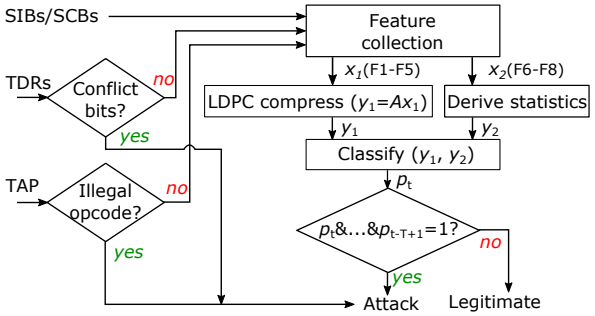


Fig. 4. The overall flow of IJTAG attack detection.

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \tag{3}$$

To construct A, two variables, namely the reduced dimension (m) and the number of ones in each column of A (d), need to be determined. Because the complexity of the original data is reflected by its density (number of non-zero entries), the optimal value for m might also be close to the average density of the original data. The number of ones in each column of A, d, reflecting the density of A, affects the orthogonality of A. According to the analysis in [19], as d increases (g decreases as a result), the correlation between the columns of A decreases when $g > 4$, and increases again when $g = 4$. Thus, the largest d that satisfies $g > 4$ is chosen (note that this value is close to but may not be the theoretically-optimal value).

C. Overall flow

Fig. 4 shows the overall flow of IJTAG attack detection. During an UPDATE-DR state, the bits in each SIB/SCB and each TDR are collected for primary checks. The user is labeled as an attacker immediately if an illegal opcode (not correspond to any JTAG function) is loaded into the IR or conflicting bits in TDRs are detected. If these checks do not detect an attack, then the features described in Table I are collected. The dimension of the collected data is

$$s = (N_S + 2 + 2N_T) \cdot w + 3 \cdot w \tag{4}$$

The first term in equation (4) corresponds to features F1-F5 (as shown in Table I) that can be reduced using an LDPC matrix due to their sparsity. The second term corresponds to features F6-F8 whose dimension can be reduced by deriving their statistics (such as max, min, and/or mean) within a window. Thus, the reduced dimension is

$$s' = m \cdot w + 3 \cdot r \qquad (1 \le r \le 3) \tag{5}$$

where r is a constant between 1 and 3 because F6-F8 may only use partial, but not all, the statistics of max, min, and mean. According to equation (5), the dimension of the reduced data is bounded because neither term depends on the number of SIBs/SCBs and TDRs. The reduced data are then supplied to a pre-trained random forest classifier that identifies the operation as either legitimate or attack. A random forest, consisting of an ensemble of decision trees, is a preferred model for classifying high-dimensional data [21].

Due to the variance that naturally occurs within IJTAG operation, the labeling of the user is delayed until sufficient evidence is collected. More precisely, only when T consecutive predictions indicate the presence of an attacker, then the user is labeled as an attacker. Upon detection of an attack, the access to the IJTAG can be restricted or obfuscated (using the technique in [22] for example), which, however, cannot be reversed by a system reset.

IV. Experiment

This section evaluates the performance of IJTAG attack detection and the LDPC-based compressed sensing. The evaluation is based on a modified version of the OpenSPARC T2.

A. Modification of OpenSPARC T2

The OpenSPARC T2 processor is used as the platform[2] for experiments [23]. The JTAG of the OpenSPARC T2 not only has access to 32 scan chains, but also can be used for dozens of testing/debugging functions. According to these functions, the OpenSPARC T2 is partitioned into 11 sub-systems, and then each sub-system is wrapped by a TDR by learning from industrial examples [16]. Then, a set of SIBs are inserted such that each wrapped sub-system is gated by a SIB. All SIBs comprise a daisy chain that is accessible via the JTAG port.

To operate the modified IJTAG, a set of legitimate operations (135 programs) are created based on the documentation of the OpenSPARC T2. Each program achieves a basic operation, like reading specific cache lines. Independent operation and pipelining operation allowed by the IJTAG are also exploited when creating the programs. The number of TDR bits that are operated simultaneously (also named degree of parallelism) varies from two to four, depending on specific scenarios. In addition to legitimate IJTAG operation, a variety of attacks (156 programs) are also created, based on the strategies described in Section II, namely reverse engineering of the SIB network and the meaning of TDR bits. Note that parallel execution can also be exploited by an attacker, but

[2]Nevertheless, the proposed detector is generic, and can be applied to other IJTAG architectures.

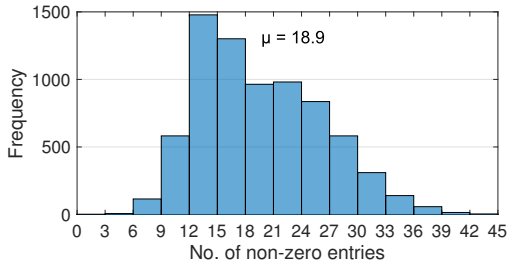


Fig. 5. The density distribution (i.e., number of non-zero entries) for the collected data whose dimension is 280.

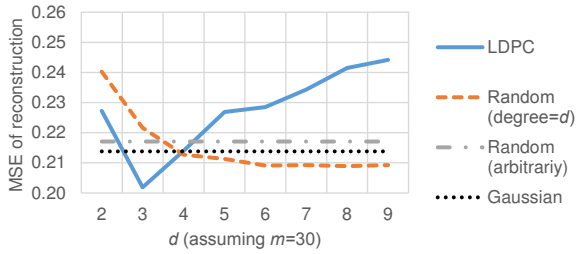


Fig. 6. The performance of reconstructing the original data is evaluated for LDPC matrices and three other types of matrices.

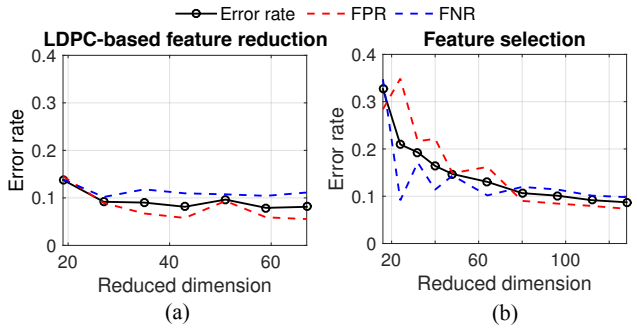


Fig. 7. The data, whose dimension is reduced by (a) the LDPC-based feature reduction and (b) feature selection using a decision tree, are classified using a random forest.

the degree of parallelism may vary in a much larger range assuming that the attacker can search many more combinations of the TDR bits. The created programs are then simulated using the modified OpenSPARC T2 and the features (as described in Table I) are collected using a sliding window of eight cycles. The density of the collected data, as shown in Fig. 5, reveals that most data have fewer than 40 non-zero entries (the dimension of the data corresponding to F1-F5 is 280). Thus, the initial assumption of sparse data is reasonable.

B. Evaluation of LDPC matrices

The collected data corresponding to F1-F5 are then reduced using an LDPC matrix. To construct an LDPC matrix A, two variables, namely m (the reduced dimension) and d (the number of ones in each column of A), need to be determined. According to the analysis in Section III-B, the largest d that satisfies $g > 4$ is preferred, and this value, through calculation, is three (assuming $m = 30$). This value is verified through simulation. Specifically, the collected data, both legitimate and attack, are compressed and reconstructed using LDPC matrices with different values of d, and then the mean squared error (MSE) of the reconstructed data is evaluated. The simulation result, as shown in Fig. 6, verifies the calculated value for m (i.e., 3), and demonstrates that quality of the reduction becomes worse for a large or small d. Fig. 6 also compares LDPC matrices to three baseline matrices, namely random matrices with degree $= d$ (each column has d ones whose positions however are random), arbitrarily-random matrices (each column has arbitrary number of ones) and Gaussian matrices (each entry is a Gaussian i.i.d random variable). According to the comparison, an LDPC matrix outperforms the other types of matrices when $d = 3$ but not for other values. Note that a random matrix with degree $= d$, showing a convergence when $d > 5$, also achieves competitive MSE, which however is still inferior to an LDPC matrix with $d = 3$.

The simulation result in Fig. 6 also verifies the possibility of estimating the optimal values for m and d without simulation, although the values vary for different systems.

C. Evaluation of IJTAG attack detection

After reducing the data corresponding to F1-F5 and deriving the statistics from the data corresponding to F6-F8 (only max is derived in this experiment), the data is then supplied to a random forest with three trees for evaluation. The evaluation exploits a five-fold cross-validation. The LDPC-based feature reduction is compared with a baseline approach, namely feature selection using a decision tree. More precisely, the features that demonstrate superior capacity of reducing the impurity of the data are selected. The performance of both approaches is measured by error rate, false positive rate (FPR, probability that a legitimate user is classified as an attacker), and false negative rate (FNR, probability that an attacker is classified as legitimate), as shown in Fig. 7. According to the results in Fig. 7a, the features reduced by an LDPC matrix demonstrates similar error rate as the original features (0.084), for a reduced dimension larger than 27. Let the reduced dimension be 27 (meaning that $m \cdot w = 24$), the LDPC matrix reduces the original dimension (304) by 91% without weakening the performance of classification. It is worth noting that $m \cdot w = 24$ is a little larger than the average density of the original data (as shown in Fig. 5). The feature selection, as shown in Fig. 7b, is also effective for dimension reduction, but is less efficient. In other words, to achieve a similar level of error rate, the reduced dimension needs to be higher than 100, which, however, is almost four times larger than the LDPC-based feature reduction.

V. Discussion

Several issues concerning the proposed detector are discussed. First, although the modified OpenSPARC T2 only has 11 instruments, the LDPC-based feature reduction is effective for a system with more instruments. This is because the reduced dimension depends more on the sparsity of the data than the number of instruments. However, more instruments may incur wiring overhead since the SIB/TDR values need to

TABLE II
SYNTHESIS RESULTS ARE COMPARED FOR THE RANDOM FOREST (WITH THREE TREES) DETECTORS.

	Original detector	w/ feature selection	w/ LDPC
Data dimension	304	96	27
Area (gate equivalent)	20,585	14,384	11,749
Compare to area of JTAG/IJTAG	14%	9.8%	8%
Latency (clock cycles)	4	4	8

TABLE III
THE PROPOSED APPROACH IS COMPARED TO PRIOR WORK.

Technique	Security	Overhead	Drawback
Access restriction [7]	+++	++	Hinder in-field test/debug
SIB locking [8]–[10]	++	+	Key leakage during distribution or via power analysis
Challenge-response [12]	+++	++	Require availability and security of a network
Attack detection	++	++	False positives and false negatives

be tapped to a global detector. This will be analyzed in future work.

Second, a comprehensive security analysis should take into consideration all types of attacks, which is difficult since different types of attacks are likely always arising. Nevertheless, the experiments in [15] show that a machine learning based detector has potential to detect unseen and disguised attacks. A possible metric for evaluating the security of the proposed detector is the open-set risk described in [24], given that IJTAG attacks are an open set (i.e., more attacks may arise).

Third, the LDPC-based feature reduction, although requiring additional adders, reduces the size of registers storing features and the depth of each decision tree. As shown in Table II, a random forest detector with LDPC-based feature reduction adds only 8% chip area compared to the JTAG and IJTAG, which is smaller than the technique using feature selection. As for latency, each row of the LDPC matrix has $\frac{n \cdot d}{m}$ ones on average, meaning that $\frac{n \cdot d}{m}$ features, on average, need to be added. This consumes $\log_4 \lceil \frac{n \cdot d}{m} \rceil$ clock cycles assuming that the adder in each clock cycle has at most four inputs. Since the LDPC matrix has m rows, the additions described above need to be operated for m times. This results in $\log_4 \lceil \frac{n \cdot d}{m} \rceil + m - 1$ clock cycles totally if the m additions are executed in a pipelining manner.

Fourth, to test the detector, a separate JTAG instruction can be defined that selects the scan chain(s) within the detector. The detector monitors the operation of the JTAG except when this instruction is executed.

VI. CONCLUSION

In this work, we propose to detect illegitimate access to IJTAG-compliant systems using a machine learning based detector. According to experiments based on the modified OpenSPARC T2, the on-chip detector adds moderate overhead ($\sim$ 8%) to the IJTAG. Further, the use of an LDPC-based feature reduction eliminates 91% of the features, and reduces circuit size by 43% without affecting detection accuracy. Nevertheless, it is hard to compare the proposed detector to other techniques shown in Table III because there is no universal metric to evaluate the level of security each method provides. However, these techniques might be combined to achieve complementing protection to the IJTAG.

REFERENCES

[1] "IEEE 1687 standard for access and control of instrumentation embedded within a semiconductor device," 2014.

[2] B. Yang, K. Wu, and R. Karri, "Scan-based side-channel attack on dedicated hardware implementations of data encryption standard," in *International Test Conference*, 2004.

[3] S. Skorobogatov and C. Woods, "Breakthrough silicon scanning discovers backdoor in military chip," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2012, pp. 23–40.

[4] Z. Basnight, J. Butts, J. Lopez, and T. Dube, "Firmware modification attacks on programmable logic controllers," *Critical Infrastructure Protection*, vol. 6, no. 2, pp. 76–84, 2013.

[5] I. Breeuwsma, "Forensic imaging of embedded systems using JTAG (boundary-scan)," *Digital Investigation*, vol. 3, no. 1, pp. 32–42, 2006.

[6] "What is JTAG?" http://www.corelis.com/education/What-Is-JTAG.htm.

[7] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Securing access to reconfigurable scan networks," in *Asian Test Symposium*. IEEE, 2013.

[8] J. Dworak and A. Crouch, "Don't forget to lock your SIB: hiding instruments using P1687," in *International Test Conference*, 2013.

[9] H. Liu and V. D. Agrawal, "Securing IEEE 1687-2014 standard instrumentation access by LFSR key," in *Asian Test Symposium*. IEEE, 2015.

[10] A. Zygmontowicz, J. Dworak, A. Crouch, and J. Potter, "Making it harder to unlock an LSIB: honeytraps and misdirection in a P1687 network," in *Design, Automation and Test in Europe*, 2014.

[11] S. Gupta, J. Dworak, D. Engels, and A. Crouch, "Mitigating simple power analysis attacks on LSIB key logic," in *North Atlantic Test Workshop*. IEEE, 2017.

[12] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, "Fine-grained access management in reconfigurable scan networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 6, pp. 937–946, 2015.

[13] S. Kan, J. Dworak, and J. G. Dunham, "Echeloned IJTAG data protection," in *Asian Hardware-Oriented Security and Trust*. IEEE, 2016.

[14] X. Ren, V. G. Tavares, and R. D. Blanton, "Detection of illegitimate access to JTAG via statistical learning in chip," in *Design, Automation and Test in Europe*, 2015.

[15] X. Ren, R. D. Blanton, and V. G. Tavares, "A learning-based approach to secure JTAG against unseen scan-based attacks," in *IEEE Computer Society Annual Symposium on VLSI*, 2016.

[16] *Avago IJTAG implementation at the IP level*, 2012.

[17] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[18] B. Bah and J. Tanner, "Improved bounds on restricted isometry constants for Gaussian matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 5, pp. 2882–2898, 2010.

[19] W. Lu, K. Kpalma, and J. Ronsin, "Sparse binary matrices of LDPC codes for compressed sensing," in *Data Compression Conference*, 2012.

[20] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[21] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[22] X. Ren, F. P. Torres, R. D. Blanton, and V. G. Tavares, "IC protection against JTAG-based attacks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018, Accepted for publish.

[23] "OpenSPARC T2," http://www.oracle.com/technetwork/systems/opensparc/, Oracle.

[24] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1757–1772, 2013.