Deep Reinforcement Learning for Resource Protection and Real-time Detection in IoT Environment

Wei Liang, Weihong Huang, Jing Long, Ke Zhang, Kuan-Ching Li *Senior Member, IEEE* and Dafang Zhang

***Abstract*—With the fast advancements of electronic chip technologies in the Internet of Things (IoT), it is urgent to address the copyright protection issue of intellectual property (IP) circuit resources of the electronic devices in IoT environments. In this work, a fast deep reinforcement learning (DRL)-based detection algorithm for virtual IP watermarks is proposed, by combining the technologies of mapping function and DRL to preprocess the ownership information of the IP circuit resource. The deep Q-learning(DQN) algorithm is used to generate the watermarked positions adaptively, making the watermarked positions secure yet close to the original design, turning the watermarked positions secure. An artificial neural network (ANN) algorithm is utilized for training the position distance characteristic vectors of the IP circuit, in which the characteristic function of the virtual position for IP watermark is generated after training. In IP ownership verification, the DRL model can quickly locate the range of virtual watermark positions. With the characteristic values of the virtual positions in each lookup table (LUT) area and surrounding areas, the mapping position relationship can be calculated in a supervised manner in the neural network, as the algorithm realizes the fast location of the real ownership information in an IP circuit. Experimental results show that the proposed algorithm can effectively improve the speed of watermark detection as also reducing the resource overhead. Besides, it also achieves excellent performance in security.**

***Index Terms*—IoT, IP watermark, deep reinforcement learning, virtual position, fast detection**

I. Introduction

THE technical requirements of intelligent manufacturing in Internet-of-Things (IoT) environments increasingly grow in astonishing speed[1], [2]. Many secure Technologies including data transmission, data protection or attack detection are proposed to ensure security of IoT[3]. Intellectual property (IP) reuse technology is critical in the development of intelligent manufacturing. The secure protection of IP circuit resource is also widely used in IP design, such as IP watermarking. However, traditional IP watermarking technologies cannot satisfy the requirements of intelligent manufacturing in terms of security and real-time performance [4], [5]. IP watermarking protection is a technology that securely hides the ownership information in an IP circuit and realizes fast detection [6], [7], [8], [9]. This technology can rapidly detect hidden watermarks and authenticate the original copyright of an IP circuit in the IoT environment at minimum costs. On the one hand, IP protection and detection technology can detect the existence of legal secret ownership information to verify the legality of hidden information; on the other hand, the technology can measure the security and robustness of the IP protection method and facilitate the efficiency of IP protection technology in the IoT environment.

In the intelligent manufacturing of IoT environments, a series of technologies should be provided to ensure the secuity[10], [11]. Especially, when disputes on IP circuit devices occur, realizing the protection and detection of IP watermark is critical. Traditional IP detection algorithms should provide the real positions of IP watermarks in verification, since such a process causes severe threats to watermarks and effective verification costs[12], [13]. In [14], the authors proposed a zero-knowledge proof-based IP watermark detection scheme, which can prove IP validity and the existence of watermarks by position scrambling algorithm without the leakage of any sensitive information. The irreversibility of the scrambling algorithm divides the verification procedure into two parts, namely, IP validity and watermark existence. Both parts require queries of several rounds to ensure the security of verification. Though, this scheme causes high complexity, high cost, and security threat in practical application. In this case, many technologies have been released and made available to optimize the security of IP protection and improve the efficiency of watermark detection, including many other authentication schemes, e.g., [15], [16].

Majority of blind IP watermark detection technologies originate from the concepts in the multimedia field [17], [18], [19]. Nevertheless, significant differences exist between them due to the features of different carriers. IP watermark detection algorithm can realize real-time protection and secure authentication of IP copyright. However, it also opens a ”backdoor” for illegal users, causing a considerable security threat. Preventing

This work was supported in part by the National Natural Science Foundation of China under Grants 61572188, 61976087, in part by the Scientific Research Program of the New Century Excellent Talents in Fujian Province University, Fujian Provincial Natural Science Foundation of China under Grant 2018J01570, and Hunan Provincial Science & Technology Project Foundation under Grant 2018TP1018.

W. Liang is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail:idlink@163.com)(Correspongding Author)

W. Huang is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China

J. Long is with Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing, Hunan Normal University, Changsha 410081, China

K. Zhang is with the Department of School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

K.-C. Li is with the Department of Computer Science and Information Engineering, Providence University, Taichung 43301, Taiwan

D. Zhang is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China

unauthorized users to capture sensitive circuit information with the backdoor is of vital importance.

II. Related Work

In recent years, the reinforcement learning (RL) algorithm is rapidly developed, which can search the optimal control policy by continuous trial and error. The development of the deep learning and the deep neural network accelerated the progress of deep reinforcement learning (DRL) [20], [21], which can address lots of practical issues, such as natural language processing, industrial applications, communications, and networking. DRL algorithm combines the perception ability of deep learning and decision-making ability of reinforcement learning, as it can learn the control policy from the original high-dimensional data. Therefore, DRL is an artificial intelligence approach that is closer to the human mind. Researches on the DRL algorithm have attracted broad considerations, involving deep Q-learning network(DQN) and the policy gradient-based DRL.

The DeepMind firstly proposed DQN algorithm in 2013 [22] and largely improved in 2015 [23]. DQN combined the convolutional neural networks and Q-learning. The neural network is an approximator of Q-function, and the weighted value is updated by using the random gradient descent method. The experience playback mechanism is utilized to make the computer learn the control policy from the high-dimensional input data. Additionally, DQN can establish two neural networks with the same structure: one is used for real-time updates, and another one updates the synchronous parameter in each time interval, which is a benefit for the convergence ability of the algorithm. So thus, this work utilized the DQN algorithm for the automatical watermark embedding and extraction decision.

The detection model of IP circuit resources can be classified into two categories, the constraint model and the unconstraint model. The former is utilized to determine the watermark positions by the artificially designed rules or the constraint model learned from the sample data, so the IP circuit can fully use the existing knowledge and avoid blind trials for the watermark positions. In this way, the learning efficiency of position detection is improved at an earlier stage. However, the constraint model cannot consider all probable cases in a complex scene, and thus the learning effect will not be satisfied. That is, even though the constraint model can improve the learning efficiency, the generalization ability is limited, as it cannot deal with the data set out of the established model. On the other hand, when the IP circuit uses the unconstraint model, nothing is known at the initial stage. It can only learn by attempting and interacting with the environment, so the efficiency of the unconstraint model is low at the earlier stage. However, it can discover all the behaviors in theory. By comparison, the constraint model and unconstraint model can compensate each other.

To address the issues of IP protection in terms of security and real-time performance, this work examines the DRL algorithm to realize the intelligent protection and real-time detection of IP circuit resources. The DRL model is combined with the virtual position mapping function in the proposed algorithm, where the sensitive fragments in IP circuits of virtual positions can be fastly detected by optimization policy. Besides, the proposed algorithm has excellent ability in autonomous reinforcement learning that remedies the defects of existing algorithms in terms of ability against attacks and the resource overhead.

III. DQN-based Insertion and Detection Strategy

It is presented in this section the DQN algorithm and means to generate the watermark positions close to the original design, enhancing the concealment and security of the embedded watermarks. Besides, the neural network has the ability to non-linear global fitting, fault-tolerance, and self-study, as it can realize infinite approximation to the non-linear function and effectively improve the efficiency of watermark extraction and detection.

All CLB resources in the FPGA are supposed to constitute an $m \times n$ matrix $C = (C_{ij})_{m\times n}$, $1 \le i \le m$, $1 \le j \le n$, where C_{ij} represents the basic resource unit of coordinates (i, j). That is,

$$C_{ij} = \begin{cases} -1, C_{ij} \text{ is used resource;} \\ +1, C_{ij} \text{ is unused resource.} \end{cases} \tag{1}$$

Firstly, the agent randomly selects a position $C_{ij} = +1$ in C as the start. The initial state of the algorithm is denoted by S, regarded as the input training set of the neural network. At the state of S_t, the action A_t is performed by using the strategy π, producing the reward value R and the next state S_{t+1}. Herein, the current reward value of each action is determined by the density of the next position.

$$R_{ij} = \sum_{j=j-1}^{j+1} \sum_{i=i-1}^{i+1} r_{ij} + 8C_{ij} \tag{2}$$

where,

$$r_{ij} = \begin{cases} 0, C_{ij} \text{ is used resource;} \\ +1, C_{ij} \text{ is unused resource.} \end{cases} \tag{3}$$

The total reward value from a state and action from policy is given by the following Action-Value Function Q:

$$Q^{\pi}(S_t, A_t) = E(R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ...|S_t, A_t) \tag{4}$$

Iteration formula (5) is used to update the value of Q until the **end** condition is satisfied.

$$Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_{A'} Q(S', A') - Q(S, A)] \tag{5}$$

The IP circuit characteristic detection layer learns the inherent characteristic from the training data set, other than extracts characteristic explicitly. The DQN algorithm utilizes a neural network to predict the value of Q. Next, the optimal path will be learned by updating the parameters in a neural network, producing the watermark positions closer to the original resources of the design. Besides, the algorithm can also improve the speed of watermark detection. Assuming that the matrix obtained from C by embedding watermark S is C', the corresponding matrix set C_k, $1 \le k \le N$ is

scrambled by the virtual position-scrambling function. This study selects $X = C_k, Y = S, 1 \leq k \leq N$ as the input training sample set T, which is provided as the training data for the convolutional neural network. Next, it obtains the feature-mapping relationship of C' and watermark information S in a supervised manner, thereby quickly extracting the watermark information.

The circuit structure of the neural network includes the input layer and the output layer. The nodes between both layers are distributed in the hidden layer, which is responsible for calculation. The relationship between the input layer and the output layer can be defined as follows.

$$Y = W^T X \tag{6}$$

where, W^T is the weight coefficient matrix, X and Y are respectively the vectors of the input layer and the output layer. Thus, the basic neuron can be expressed as follows.

$$y = f(\sum_{i=1}^{N} \omega_i x_i + b) \tag{7}$$

where ω_i is the weight, b is an offset of the input data, and $f : R \to R$ is an activation function.

To enhance the robustness and convergence of the algorithm, two neural networks are introduced in the DQN algorithm. One is the Target-net with fixed parameters, which can generate target-$Q = R + \gamma \max_{A'} Q(S', A', \omega)$, while another one is Main-net with real-time updated parameters to generate eval-Q value, so that the Target-net synchronize the parameters with Main-net at each interval. Therefore, the loss function in training the parameters of the neural network is defined as follows.

$$L(\omega) = E[(R + \gamma \max_{A'} Q(S', A', \omega) - Q(S, A, \omega))]^2 \tag{8}$$

The random gradient descent method is utilized to update the weight value as follows.

$$\begin{aligned} \nabla_\omega L(\omega) =& E[(R + \gamma \max_{A'} Q(S', A', \omega) - Q(S, A, \omega)) \\ & \nabla_\omega Q(S, A, \omega)] \end{aligned} \tag{9}$$

IV. DRL-BASED IP WATERMARKING ALGORITHM

In IoT environments, IP watermarking algorithms have remarkable differences due to different IP carriers. In traditional IP watermarking algorithms, the signature information of the IP owner is a sequence of 0 and 1, generated by the watermark generator. These digital signals should be inserted into the IP design without affecting the standard functionality of the circuit. The watermarked design can be used for ownership verification, while the watermark detector is utilized to detect the existence of watermarks. These algorithms have enormous hardware costs, and real-time performance is non-optimal.

As shown in Fig. 1, the proposed algorithm utilizes the DQN model in the DRL algorithm to train the characteristic information of watermark positions, and the virtual mapping method is used to protect the real watermark positions. The design of the DQN-based IP-watermarking algorithm includes three parts, namely, virtual watermark generator, virtual watermark embedder, and virtual watermark extractor. Firstly, the signature of the IP owner is preprocessed, so the characteristic information of watermark positions is trained using DQN. Next, the watermarks are mapped to the virtual watermark positions so that when the virtual watermark information is verified, the watermark detector map to the real watermark positions and detects the watermarks in bitfile to realize the fast detection of IP ownership information.

A. Watermark Generation

The IP ownership information includes the signature of the IP owner, product brand, among other featured information. As shown in Fig. 2 and encrypted by the AES algorithm, the signature of the IP owner Sig produces the ciphertext $Csig$ under the control of secret key key. The scrambling algorithm is used to scramble $Csig$ with key $Skey$, so the generated sequence is divided into L fragments $Wsig$ for embedding. As the next step, insert the watermark fragments into the IP design without affecting the standard functionality of the IP circuit. That is, the concrete steps of the watermark generation are detailed as follows.

Step1: The brand or signature of IP circuit Sig is inputted into the watermark generator, and represented by a binary sequence denoted by $S = b_1, b_2, ..., b_m$, $b_i = 0$ or 1,

Step2: S is encrypted by AES after adding the time stamp, and the ciphertext $Csig$ is generated,

Step3: Based on the generation of virtual random watermark positions, the ciphertext is divided into watermark L fragments $Wsig$, as also the corresponding random threshold value of virtual positions are generated,

Step4: The virtual watermarks are generated for embedding, and they have high randomness.

B. Watermark Embedding

The IP design at the physical level always generates the bitfile of the IP circuit. It is aimed at this work the identification of unused resources in used slices for watermark embedding. DRL technology is utilized to realize virtual watermark embedding, and the watermarks should be initially transformed into binary code to improve the embedding capacity and reduce additional embedding costs, thereby decreasing the performance effect on IP design. The flow of the virtual watermark embedding is depicted in Fig. 3, .

Initially, the IP ownership information is encrypted then transformed into a linear combination of watermark positions and actual watermark bits. The watermark positions are regarded as additional constraints, and watermark bits are inserted into the design. In this case, the use of minimally inserted information denotes important ownership information to improve watermark capacity. The concrete steps are as follows.

Step1: DQN training. The agent utilizes the DQN algorithm to search for the watermark resources. The Q value updating function $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_{A'} Q(S', A') - Q(S, A)]$ is utilized to generate each state behavior pair (S, A)

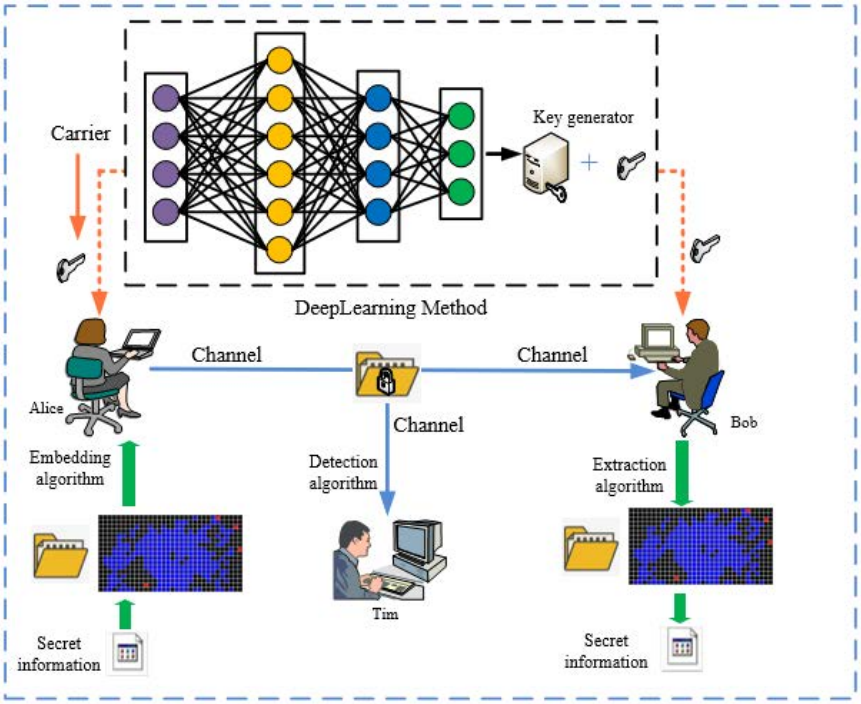


Fig. 1: Flow of IP circuit protection

until producing the target function. Finally, N positions x_i, y_i closest to the originally used resources are generated for watermark embedding, and such a rule is as follows.

$$C'_{x_i y_i} = C_{x_i y_i} + b_i \tag{10}$$

The transformed matrix is denoted by C'.

Step2: Virtual position generation. A reversible position scrambling function $f((x,y),b) : (x,y) \rightarrow (x',y')$ is constructed. Hereby, b=0 or 1 is the watermark at position (x,y) that satisfies the following conditions.

1)$l = \lfloor \frac{x+y}{2} \rfloor$;
2)$h = x - y$;
3)$h' = 2h + b$;
4)$x' = l + \lfloor \frac{h'+1}{2} \rfloor,\ y' = l - \lfloor \frac{h'}{2} \rfloor$.

The function can be proven as reversible. With x', y', the following expressions (11) and (12) can be calculated.

$$l' = \lfloor \frac{x' + y'}{2} \rfloor \tag{11}$$

$$h' = x' - y' \tag{12}$$

Then, we have (13) and (14).

$$h = \lfloor \frac{h'}{2} \rfloor \tag{13}$$

$$l = x' - \lfloor \frac{h'+1}{2} \rfloor \tag{14}$$

Accordingly,

$$l = \lfloor \frac{x+y}{2} \rfloor \tag{15}$$

$$h = x - y \tag{16}$$

With (15) and (16), (17) and (18) are deduced.

$$x = l + \lfloor \frac{h+1}{2} \rfloor \tag{17}$$

$$y = l - \lfloor \frac{h}{2} \rfloor \tag{18}$$

Therefore, $f(\bullet)$ is a reversible function with given b.

The original positions (x_i, y_i) can be mapped to (x'_i, y'_i) with the scrambling function, which scrambling iteration times k are selected to satisfy the security requirement. The virtual positions (x_i^*, y_i^*) after k rounds of scrambling are generated and the corresponding matrix is denoted by C_k. The watermarks generated in the above section can be inserted into these watermarked positions, thereby producing a watermarked IP design.

Step 3: Encryption. The generated virtual positions (x_i^*, y_i^*) are encrypted by asymmetrical encryption algorithm. The ciphertext is L, and the secret key is Key.

The pseudo-code of the watermark generation algorithm is stated as follows.

C. Watermark Resource Extraction

Watermark extraction is performed when IP disputes occur. If illegal users obtain an IP design with the virtual watermark and illegally making use of it in their system, the IP owner can apply for copyright verification. The detailed extraction process is as follows.

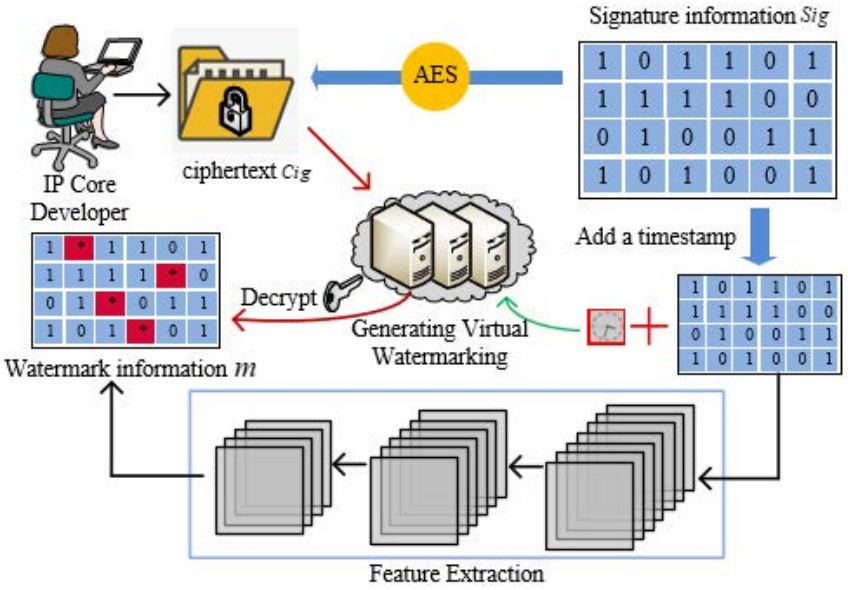


Fig. 2: Generation of virtual watermarks

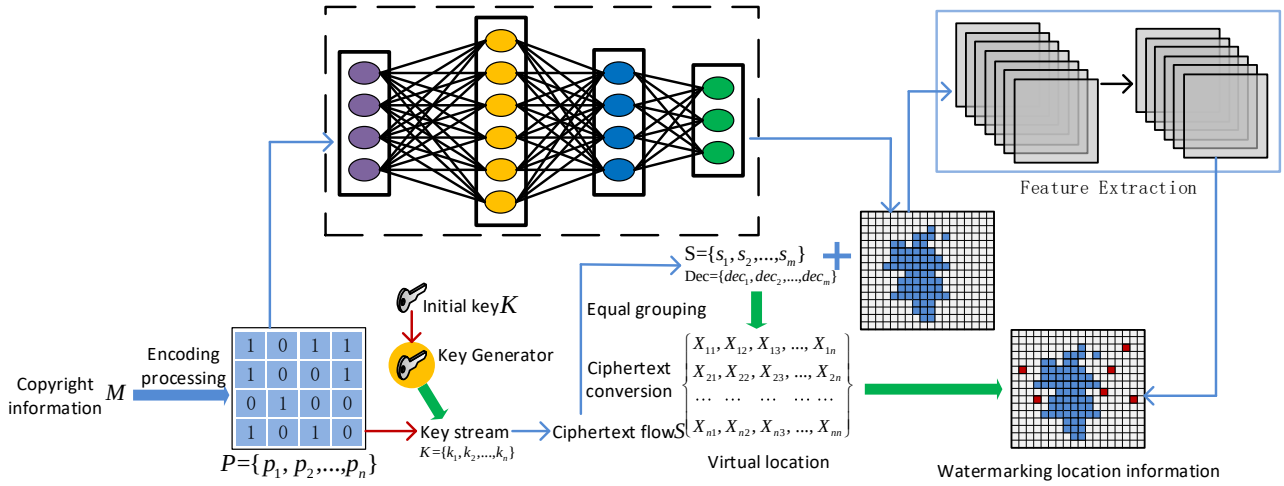


Fig. 3: Flow of virtual watermark embedding

(1) The IP owner uses his private key *Key* to decrypt the ciphertext L and acquire the virtual positions (x_i^*, y_i^*), which can be reversibly mapped to the original watermark positions (x_i, y_i) with f^{-1},

(2) All elements in matrix C' that correspond to position information (x_i, y_i) are decreased by 1. The transformed matrix is denoted by C''. Still, the agent utilizes the DQN algorithm to randomly select the position with $C_{ij} = +1$ in C'' as the start point, with S denoting the initial state of the algorithm and regarded as the input training set of the neural network. At the state S_t, the strategy π' that moves to position $C_{ij} = 0$ is selected to perform action A_t, producing the reward value R and the next state S_{t+1}. It is repeated until the binary sequence b_i is generated and transformed into ciphertext.

(3) The key $K = k_1, k_2, ..., k_n$ can be used to decrypt the ciphertext and further retrieve the original ownership information.

The digital virtual IP watermark extraction is to extract the embedded watermarks from the watermarked design. The DRL model can rapidly locate the position range of virtual watermarks and calculate the position mapping relationship in a supervised manner, so thus the actual ownership information can be extracted from the watermarked IP design. In this research, the inputs of the watermark extractor are the watermarked IP design and the key file that records the LUT positions and time of watermark embedding. DQN is used

Algorithm 1 Watermark generation algorithm.

Input:
Original IP core C, watermark $S = b_1, b_2, ..., b_m$;
Output:
Virtual watermark C_k ;
Repeat each episode;
Agent utilizes DQN algorithm to explorer the environment and updates Q function;
Utilize Q function to perform the next episode;
Generate the optimal initial coordinates (x_i, y_i) of positions;
$C'_{x_i y_i} = C_{x_i y_i} + b_i$;
$C = C'$;
for $j = 1$ to k **do**
  for $i = 1$ to m **do**
    $(x_i^{(j)}, y_i^{(j)}) = f((x_i^{(j-1)}, y_i^{(j-1)})$;
  end for
  $C = C_k$;
end for
return C_k.

to generate the classification function, so the key file can be used to retrieve the LUT position information next. Finally, the virtual watermarks can be extracted from the watermarked bitfile, as depicted in Fig. 4.

The pseudo-code of the watermark extraction algorithm is illustrated as follows.

Algorithm 2 Watermark extraction algorithm.

Input:
(x_i^*, y_i^*), C';
Output:
Watermark b_i ;
for $i = 1$ to m **do**
  for $j = k$ to 1 **do**
    $(x_i^{(j)}, y_i^{(j)}) = f^{-1}(x_i^*, y_i^*)$;
    $C'_{x_i y_i} = C'_{x_i y_i} - 1$;
  end for
end for
Input $x_k = (x_i, y_i), y_k = C'_{x_i y_i} - 1$;
Use DQN algorithm to get the position classification function $y = f(x)$;
for $i = 1$ to m **do**
  $b_i = y_i$;
end for
return b_i.

D. DRL Watermark Resource Detection

In IoT environments, the authentication parties are assumed to include a prover and a verifier, so three shared parameters (α, t, n) should be initialized. In the detection procedure, the selection of a large prime number n and production number α in the detection procedure may cost considerable time overhead in the calculation. The embedded information generates the parameter t, so the protected virtual watermark positions will not be leaked to the verifier in the proposed scheme. If the prover requires to demonstrate the ownership to the verifier, the public key parameter t should be generated and sent next to the verifier at the server for verification purposes. In practical ownership verification, the shared parameters should be generated before IP verification.

The secure IP authentication protocol adds identity authentication of the virtual watermark in real-time detection. The algorithm has the functional nonlinear global fitting capacity and adaptive learning capacity due to the use of the convolutional neural network in DRL. On this basis, the designed mapping function can effectively improve the detection efficiency of virtual watermarks, and the detection steps are as follows.

Step1: The prover selects a random number $r \in 2, ..., 2^j$, $2^j < n$, and calculates $x = \alpha^r \mod n$. The request information x is sent to the verifier via the communication connection in advance,

Step2: The verifier selects a random number $k \in 2^8 0, ..., 2^j$ after receiving the information from the prover, and sent to the prover via the communication connection,

Step3: The prover receives k and extracts m from the watermarked IP core, and a random number r is then selected. With m, r, k and n, $y \equiv km + r(\mod \varphi(n))$ can be calculated, and the results are sent to the verifier next,

Step4: The verifier receives y and uses the given parameters x, y, k and the public parameter to verify whether $x = t^{-k}\alpha^y \mod n$ is satisfied,

Step5: All CLB resources in FPGA are assumed to construct an initial matrix $C = (C_{ij})_{m \times n}$. The original watermark positions (x_i, y_i) can be mapped to the virtual positions (x_i^*, y_i^*) after k rounds of scrambling, so the newly generated matrix is denoted by C_k. The training sample set is $T = (X_1, Y_1), (X_2, Y_2), ..., (X_N, Y_N)$, $X_k = C_k$, $Y_k = S_k$, $1 \leq k \leq N$, while T provides training data for the convolutional neural network, and then generates a mapping relationship between C and S supervised manner, thereby realizing the watermark detection.

The pseudo-code of the virtual watermark detection algorithm is depicted as follows.

Algorithm 3 Watermark detection algorithm.

Input:
Training sample set T, test set C;
Output:
Watermark S ;
Extract the characteristics of the training sample set $SIFT(T)$;
for $i = 1$ to N **do**
  $\hat{Y} = F_n(...(F_2(F_1(X_i W_1)W_2))...W_n)$;
  $\min \frac{1}{2}(y_i - \hat{y}_i)^2$;
end for
Input C;
return $S = b_1, b_2, ..., b_m$.

V. Performance Analysis

In IoT environments, the use of the neural networks in DRL can securely hide the watermarks into random virtual

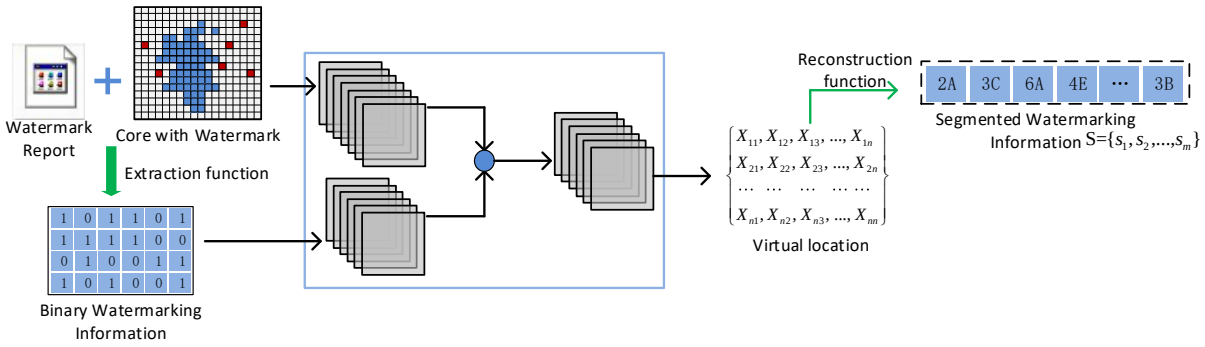


Fig. 4: Flow of watermark extraction

positions of low-level IP circuits. Besides, it can fast detect the ownership information of IP design. In this section, we analyze the performance of the proposed algorithm in terms of reliability, transparency, and performance overhead.

A. Resistance against Attacks

This algorithm utilizes the feature of DRL technology that can make fast decisions and optimization for sensitive information, since the security is verified in cryptography and the detection system includes software and hardware running on the network. For example, several hardware attack methods may make the IP detection system run out of the resources, as it causes low detection speed. Through performance evaluation experiments, the IP watermark detection system initially captures the watermark positions from the IP design. The data-capturing module runs at the physical level that realizes data capturing and storage stably. After that, the detection system does not crash after suffering from attacks. The watermarked IP design has a minimal probability of being the same as the non-watermarked design. The larger the watermark capacity is, the smaller the collision probability will be; therefore, the reliability is higher. This algorithm utilizes deep-learning-based IP watermark detection, which is suitable for IP copyright authentication. In this case, the collision probability after embedding becomes smaller, and therefore, the reliability becomes higher.

B. Overhead Analysis

The overhead during the operation includes the resource occupation and power overhead after watermark embedding. That is, the algorithm utilizes the unused resources in the original design that cause the growth of the resource occupation. There exist numerous programmable resources in the FPGA device that can be used for watermark insertion, though the additional resources caused by the watermark embedding can be ignored. The unused resources in the original design are used for watermark embedding, so the watermarked resources are not activated, and therefore, the watermark embedding does not generate additional power overhead, so the algorithm has the feature of zero power overhead.

C. Complexity Analysis

In this section, the complexity of effective DRL-based detection of the virtual mapping positions is analyzed, as the conditions depend on the preset selected characteristic value. If the DRL-based selection scheme is (t, n), the algorithm is calculated by n times. That is, the watermark fragments are used by the algorithm for n times. Assuming that the total length of IP watermark is k, and the frequency of IP watermark used by the algorithm is kn, the time complexity of this algorithm is $O(n^2)$. Additionally, the storage space as $k(n+1)$ is calculated and the space complexity is $S(n^2)$.

VI. Experimental Result and Analysis

The IP cores in this experiment come from `opencores.org` [24], and the FPGA type covers all models of Xilinx Virtex-II family devices. In this section, we use Xilinx Virtex-II XC2V500 FPGA for evaluation. The IP design is implemented by the ISE tool to generate the bitfile at the physical level used as the watermark carrier, while the system is realized by Java language. The computing server must have a JDK environment and Java compiler, such as Elipse.

Many metrics, such as robustness, reliability, and security, can be used to evaluate the performance of the IP-watermarking system. The security of virtual mapping watermarks is based on the protection of IP watermarks. However, IP verification security depends on discrete logarithm computation since the watermarks are inserted into the bitfile at the physical level. If the watermark positions in the key file are correct, then the watermarks can be detected and extracted. We primarily evaluate the performance in aggregation degree of watermark positions, security, detection speed, and other additional overhead.

A. Aggregation Degree of Watermark Positions

As an IP core implemented on different chip types achieves different resource occupations, the testing results on the implementation of IP benchmarks on Xilinx Virtex-II XC2V500 FPGA vary. However, the occupied resources are the same

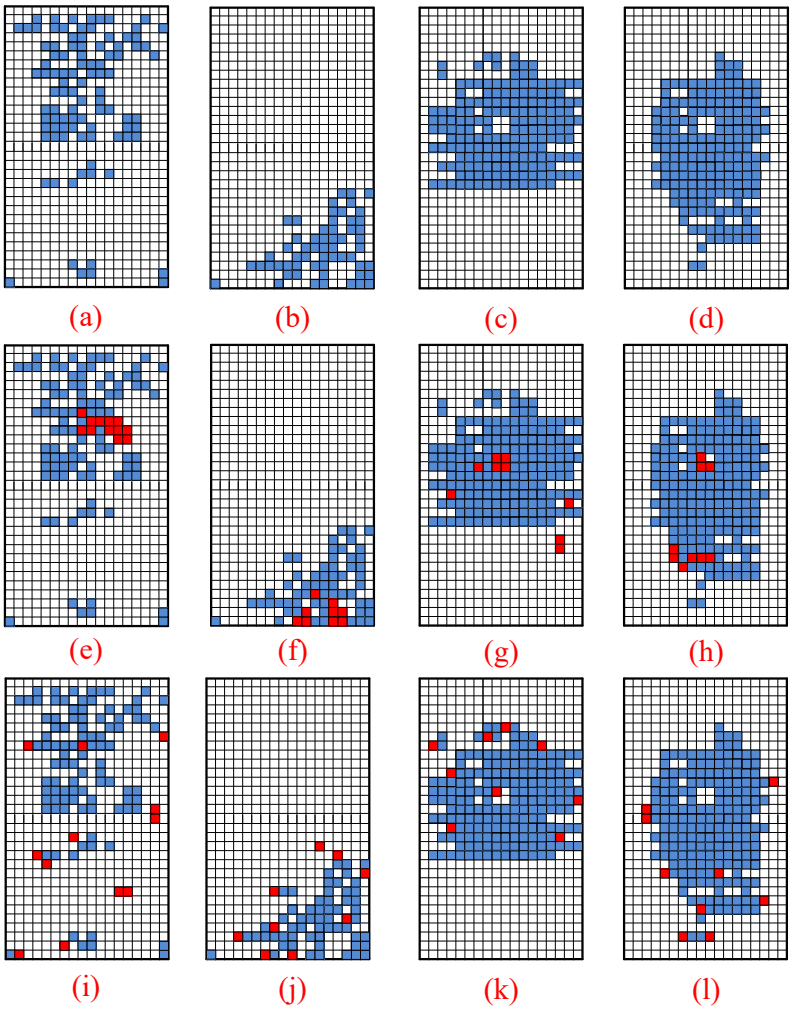


Fig. 5: The CLB resource occupation of IP core

as realizing the same IP core. The resources before and after embedding 128 bits watermark are shown in Fig. 5.

The resource distribution of different IP cores on XC2V500 FPGA is shown in Figs. 5(a)-5(d). As depicted in Figs. 5(e)-5(h), the aggregation degree is small after watermark embedding. However, the watermark embedding based on DRL achieves a high aggregation degree and low watermark detection overhead, as shown in Figs. 5(i)-5(l).

The watermark detection algorithms in [25], [26] improve the speed to crack hash function. Nonetheless, the 128-bit hash message is challenging to obtain reversibly. Excellent security performance is achieved, though the density of resource occupation is high. In the proposed DRL based algorithm, the mapping function and DRL are combined. That is, the ANN algorithm is utilized for training the characteristic vectors of the IP circuit, so the trained characteristic function of virtual watermark positions is generated. The generated virtual watermarks are calculated by the hash algorithm and compressed for embedding. In this case, the resource occupation after DRL based watermark embedding is reduced.

As shown in Fig. 6, the proposed watermarking algorithm is assumed to have a constant watermark capacity. With an increase in the occupied resources of the IP circuit, the virtual watermarks cause considerable overhead.

B. Detection Speed

The occupied resources in IP design increase with the IP watermark capacity, given that such a condition causes low detection speed and high time delay. The proposed DRL-based algorithm generates virtual watermarks and embeds them into mapped virtual positions. The watermark detection increases additional resource overhead, as time delay also occurs in practical detection. The speed ratio of the three algorithms is shown in Fig. 7(a), where we observe that the proposed DRL-based algorithm has better improvements in detection speed than other algorithms since the real watermark needs not to be detected though undergoes real-time verification. It is noted that the detection speed is improved, though the detection delay is not the best one, as shown accordingly to Fig. 7(b).

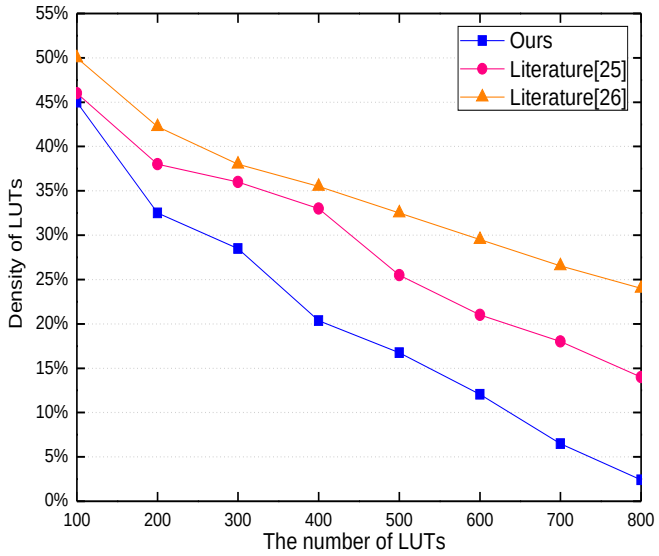
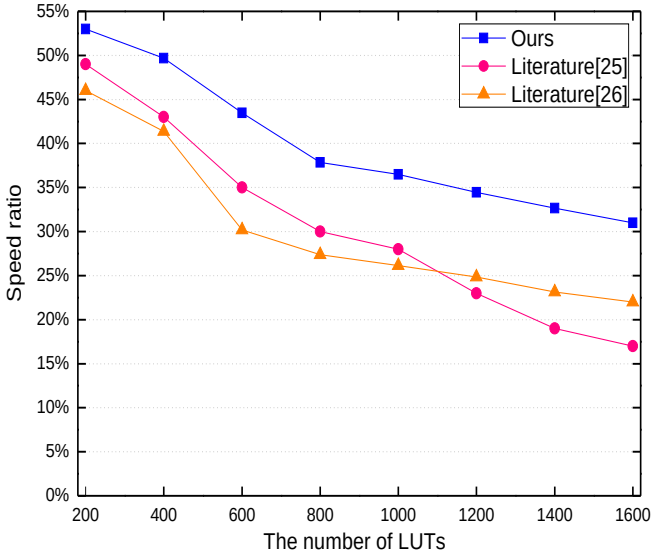
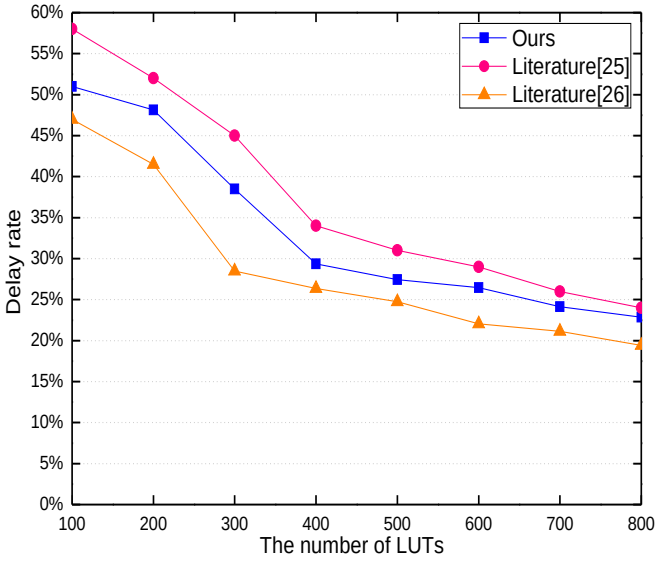


Fig. 6: Comparison of IP virtual watermark resource occupation



(a)



(b)

Fig. 7: Comparison of watermark detection speed and time delay

C. Security

As numerous typical attacks may occur, robustness and ability against typical attacks are of great importance and considered for the evaluation. The robustness can be evaluated by the normalized correlation (NC) and bit error ratio (BER), since both metrics can be used in different scenes. BER is suitable to extract watermark, while NC can be used to determine whether the watermark exists. The proposed algorithm can detect and evaluate the security and robustness by BER, as calculated in (19).

$$BER = \frac{100}{B}\sum_{n=0}^{B-1}\begin{cases}1, w(n) \neq w'(n)\\ 0, w(n) = w'(n)\end{cases} \tag{19}$$

where B is the length of the watermark, w, and w' are the original watermark and extracted watermarks, respectively. If the watermark is ultimately detected, then the value of BER is 0; BER is 100%, if otherwise. In the proposed algorithm, the watermark length is 128 bit, and the BER is shown in Fig. 8. The X-axis is the ratio between the broken watermark bits and the total watermark bits, while the Y-axis is the value of BER. From the figure, BER is proportional to the damaged watermark bits.

D. Overhead Comparison

The proposed algorithm is compared in terms of resource occupation with the methods presented in [25], [26] and presented in Table 1. The first column lists several benchmark circuits used in the experiment, the second column refers to the appropriate FPGA devices for these benchmarks, while the third column represents the LUT occupation in the original IP core, denoted by LUT_o. The remaining columns present the comparisons in terms of LUT resources, where LUT_w is the number of used LUTs in the watermarked IP core. From Table. 1, it can be seen that the resource overhead in the proposed algorithm is superior to those in comparative algorithms.

VII. Conclusions and Future Work

In IoT environments, resource protection and real-time detection are critical to ensure the security of the IP design. In this work, a DRL-based virtual IP detection algorithm is proposed to ensure the security of low-level hardware in the intelligent manufacturing of IoT environments. The proposed algorithm combines DRL technology and virtual mapping function, yet the watermark embedding, extraction,

TABLE I: Comparison of LUT resource overhead

Benchmarks	FPGA device	LUT_o	Literature[25]		Literature[26]		Ours	
			LUT_w	LUT Overhead	LUT_w	LUT Overhead	LUT_w	LUT Overhead
analytic	XC2V250	298	325	9.060%	358	20.134%	306	2.685%
des56	XC2V500	692	735	6.213%	750	7.514%	700	1.156%
storm	XC2V1500	7305	7386	1.108%	7398	1.273%	7313	0.110%
aes	XC2V2000	1378	1435	4.136%	1452	5.370%	1386	0.581%
cpuc	XC2V3000	3415	3486	2.079%	3496	2.371%	3423	0.234%
rs_dec4	XC2V4000	25929	25998	0.226%	25993	0.247%	25937	0.031%
Average				3.803%		6.151%		0.697%

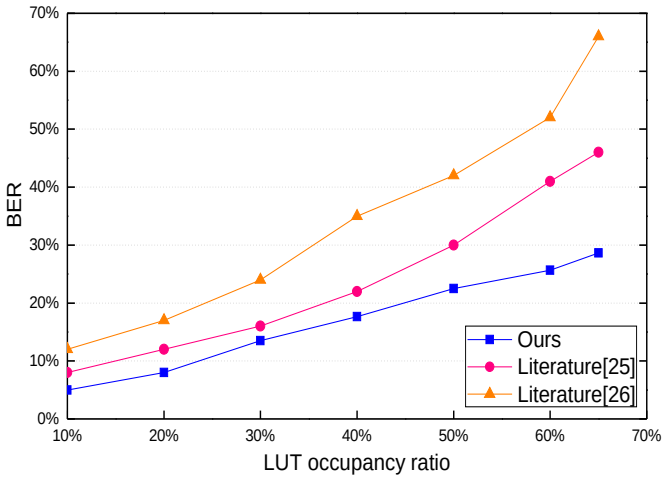


Fig. 8: The BER comparison of virtual watermark

and detection algorithms are detailed in this work. The performance of detection speed, resource overhead, and security are analyzed and compared. Experimental results show that the proposed algorithm can rapidly detect the watermark without affecting the standard functionality of the original IP design. Besides, the proposed algorithm achieves high security and low overhead performance. As future work, we step in to research a blind detection model for IP watermark based on DRL technology, as more optimization tasks can be performed to achieve better security performance and detection efficiency.

REFERENCES

[1] X. Li *et al.*, "A robust and energy efficient authentication protocol for industrial internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1606–1615, 2018.

[2] W.Liang *et al.*, "A secure fabric blockchain-based data transmission technique for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 15, pp. 3582–3592, 2019.

[3] W.Liang, K. C. Li *et al.*, "An industrial network intrusion detection algorithm based on multi-characteristic data clustering optimization model," *IEEE Transactions on Industrial Informatics*, 2019.

[4] X. Li, J. Niu, and M. Z. A. Bhuiyan, "A robust ecc based provable secure authentication protocol with privacy preserving for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3599–3609, 2018.

[5] A. Sengupta, S. Bhadauria, and S. P. Mohanty, "Embedding low cost optimal watermark during high level synthesis for reusable ip core protection," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, Montreal, 2016, pp. 974–977.

[6] J. Zhang and G. Qu, "Recent attacks and defenses on fpga-based systems," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 12, no. 3, pp. 14–38, 2019.

[7] D. Roy and A. Sengupta, "Low overhead symmetrical protection of reusable ip core using robust fingerprinting and watermarking during high level synthesis," *Future Generation Computer Systems*, vol. 71, pp. 89–101, 2017.

[8] W. Liang, W. Huang, W. Chen *et al.*, "Hausdorff distance model-based identity authentication for ip circuits in service-centric internet-of-things environment," *Sensors*, vol. 19, no. 3, pp. 487–505, 2019.

[9] J. Zhang, B. Qi, Z. Qin, and G. Qu, "Hcic: Hardware-assisted control-flow integrity checking," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 458–471, 2019.

[10] W.Liang, Y. Fan *et al.*, "Secure data storage and recovery in industrial blockchain network environments," *IEEE Transactions on Industrial Informatics*, 2020.

[11] W.Liang, J. Long *et al.*, "TBRS: A trust based recommendation scheme for complex cps network," *Future Generation Computer Systems*, pp. 383–398, 2019.

[12] A. Sengupta, D. Kachave, and D. Roy, "Low cost functional obfuscation of reusable ip cores used in ce hardware through robust locking," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 4, pp. 604–616, 2018.

[13] X. Li, J. Niu, J. Liao *et al.*, "Cryptanalysis of a dynamic identity based remote user authentication scheme with verifiable password update," *International Journal of Communication Systems*, vol. 28, no. 2, p. 374–382, 2015.

[14] J. Zhang and L. Liu, "Publicly verifiable watermarking for intellectual property protection in fpga design," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 25, no. 4, p. 1520–1527, 2017.

[15] J. Long, W. Liang, K. C. Li, D. Zhang, M. Tang, and H. Luo, "Puf-based anonymous authentication scheme for hardware devices and ips in edge computing environment," *IEEE Access*, vol. 7, no. 1, pp. 124 785–124 796, 2019.

[16] W. Liang, S. Xie, J. Long, K. C. Li, D. Zhang, and K. Li, "A double puf-based rfid identity authentication protocol in service-centric internet of things environments," *Information Sciences*, vol. 503, no. 1, pp. 129–147, 2019.

[17] A. Sengupta and S. Bhadauria, "Untrusted third party digital ip cores: Power-delay trade-off driven exploration of hardware trojan secured datapath during high level synthesis," in *Proceedings of the 25th edition on Great Lakes Symposium on VLSI. ACM*, Pittsburgh, 2015, pp. 167–172.

[18] C. Marchand, L. Bossuet, and E. Jung, "Ip watermark verification based on power consumption analysis," in *2014 27th IEEE International System-on-Chip Conference (SOCC). IEEE*, Las Vegas, 2014, pp. 330–335.

[19] M. Meenakumari and G. Athisha, "A survey on protection of fpga based ip designs," *International Journal of Advanced Electrical and Electronics Engineering*, vol. 2, no. 2, pp. 93–99, 2013.

[20] K. Zhang, Y. Zhu, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Deep learning empowered task offloading for mobile edge computing in urban informatics," *IEEE Internet of Things Journal*, vol. 1, no. 1, to be published.

[21] K. Zhang, S. Leng, X. Peng, P. L. S. Maharjan, and Y. Zhang, "Artificial intelligence inspired transmission scheduling in cognitive vehicular communications and networks," *IEEE Internet of Things*, vol. 1, no. 1, to be published.

[22] V. Mnih *et al.*, "Playing atari with deep reinforcement learning," *Nature*, 2013.

[23] V. Mnih, K. Kavukcuoglu, and D. Siler, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[24] OpenCores. (1999) Opencores web site. [Online]. Available: http://www.opencores.org

[25] J. Long, D. Zhang *et al.*, "A robust low-overhead watermarking for field authentication of intellectual property cores," *Computer Science and Information Systems*, vol. 13, no. 2, pp. 609–622, 2016.
[26] D. Saha and S. Sur-Kolay, "Secure public verification of ip marks in fpga design through a zero-knowledge protocol," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 10, pp. 1749–1757, 2012.