

RESEARCH ARTICLE

A Scalable and Accurate Chessboard-Based AMC Algorithm With Low Computing Demands

YUQIN ZHAO ^{ID}, WILLIAM C. J. GAVIN, TIANTAI DENG ^{ID}, (Member, IEEE),
EDWARD A. BALL ^{ID}, (Member, IEEE), AND LUKE SEED ^{ID}

Department of Electronic and Electrical Engineering, The University of Sheffield, S1 3JD Sheffield, U.K.

Corresponding author: Tiantai Deng (t.deng@sheffield.ac.uk)

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Early Career Researcher Studentship, which is also known as the Doctor Training Partnership (DTP) project.

ABSTRACT Automatic Modulation Classification (AMC) is a technique used to identify signal modulations in applications like IoT devices, cognitive radar, software-defined radio, and electronic warfare. These applications could be applied to IoT devices. With future wide applications of IoT devices, AMC algorithms need to be more compact yet suitable for embedded devices with limited resources and remain acceptable accuracy. Although current AMC algorithms deliver high accuracy, they require substantial computing power, making them unsuitable for IoT devices. This paper introduces the novel Chessboard-based Automatic Modulation Classification (CAMC) algorithm, which has dramatically high accuracy. Test results reveal that CAMC achieves 99%* accuracy under a 3dB SNR condition and 100% above 5dB SNR. Meanwhile, this algorithm is scalable and demands less computing power. It offers better accuracy results compared to state-of-the-art AMC algorithms, classifying mainstream modulations in IoT devices like BPSK, QPSK, 8PSK, and 16QAM, but requires less computing power than existing algorithms. Additionally, CAMC is hardware-friendly due to its inherent parallelism and scalability. The novelty of this paper is to classify 4 different modulations in a low-computation-loading required and hardware-friendly way and achieve a high accuracy of over 99%* above SNR of 3dB. (* Accuracy that most of the time could reach)

INDEX TERMS Communications technology, classification algorithms, modulation, Internet of Things, parallel algorithms, image classification, software radio, phase modulation.

I. INTRODUCTION

Automatic modulation classification (AMC) serves as an intermediary process between signal detection and demodulation, enabling the identification of signals with unknown modulations [1]. AMC finds applications in radio frequency (RF) signal processing fields, including cognitive radar and electronic warfare, as well as mobile communication systems such as software-defined radio (SDR) [2], [3]. By addressing high-density spectrum issues in communication environments, AMC enhances the efficiency of demodulating signals with unknown modulations, thereby alleviating spectrum congestion and improving overall system performance [4], [5]. Based on the advantages of AMC, now it is a trend that

AMC-related technologies are being applied to IoT devices. Due to the large quantity of IoT devices in this world, the spectrum limits problem is not ignorable. Therefore, more communication-based technologies like cognitive radar and SDR, are now applied to an IoT device [6], [7].

AMC methodologies can be broadly categorized into two main approaches: feature-based approach and decision-theoretic approach. In the feature-based approach, features are first extracted from the signals, and then the modulation is classified based on these features. This method is straightforward and does not necessitate prior knowledge of communication or modulation. On the other hand, the decision-theoretic approach demands comprehensive communication knowledge and detailed information, such as phase jitter, erroneous channel state information, and frequency offset. This method involves maximizing the

The associate editor coordinating the review of this manuscript and approving it for publication was Julien Le Kerneec ^{ID}.

likelihood information before using it to classify the modulation [2]. Recently, traditional AMC methods have been integrated with deep learning applications to achieve higher accuracy, as demonstrated in [4] and [8]. In [4], constellation and spider images serve as input for a Convolutional Neural Network (CNN) to classify the signal via the constellation image.

In recent years, Software-Defined Radio (SDR) has gained popularity, with ARM(Advanced RISC Machine)-based, DSP(Digital Signal Processor)-based, and FPGA(Field Programmable Gate Arrays)-based SDRs available. However, ARM-based and DSP-based SDRs may not meet the stringent timing requirements of future communication systems due to shrinking symbol time windows. Consequently, FPGAs, with their high sampling frequency capabilities [9], [10], could become the only viable option. Thus, it is essential to consider hardware implementation aspects of AMC algorithms during the design stage. Although the algorithm in [11] achieves nearly 100% accuracy even at -2 dB of SNR, its complexity poses challenges for FPGA implementation due to the logarithmic, root, and division operations involved.

Performing operations like logarithms on FPGAs is resource-intensive and power-consuming [12]. Additionally, limited reconfigurable hardware capability for floating-point calculations may decrease accuracy during implementation. Utilizing lookup tables can attain high accuracy but significantly increases hardware resource usage and necessitates hardware expertise [13], [14].

Deep-learning-aided methodologies can enhance accuracy but exhibit drawbacks, such as being computationally intensive, energy-consuming, and less practical in real-life scenarios [11]. Furthermore, generated features are not easily explainable, hindering optimization based on domain-specific knowledge. Existing AMC algorithms exhibit an imbalance between complexity and accuracy.

To sum up, based on the related work mentioned above, the problems that remain in the state-of-the-art designs are listed below:

- 1) Complicated calculations (decision-theoretic method) or a large number of calculations (deep-learning-aided method) are required to achieve high accuracy, which leads to a large amount of computation loading.
- 2) Algorithms mentioned are hardware-friendly and could lead to accuracy degeneration when applied to hardware.
- 3) Existing work consumes a long processing time.

With more stringent speed and accuracy requirements in future communication systems, we propose a novel, low-complexity AMC algorithm capable of accurately detecting four mainstream modulations: BPSK, QPSK, 8PSK, and 16QAM. These four modulations are chosen since they are still widely used in the IoT communication standards, including WIGHTLESS-P and WEIGHTLESS-W [15]. By choosing the feature-based method combined with a hardware-friendly algorithm, calculation loading could be

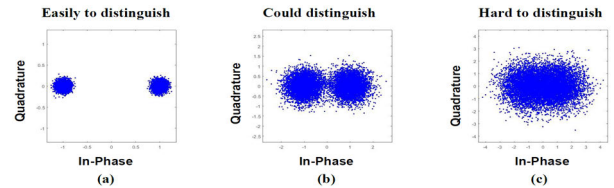


FIGURE 1. BPSK scatter plot under different SNR (a) 20 dB (b) 5 dB (c) -2 dB.

eliminated, and it could keep a better performance when applied to a hardware platform. Our inherently parallel, scalable, and hardware-friendly algorithm achieves accuracy comparable to existing methods while significantly reducing complexity. Detailed contributions are as follows:

1. We propose a Chessboard-based AMC (CAMC) algorithm which has competitive accuracy as the existing algorithms. The accuracy could reach 99% when SNR is over 3dB and 100% when SNR is over 5dB.
2. This CAMC algorithm could reach 97% accuracy under 2000 of test symbols and perform with good noise-tolerant ability. When the number of test symbols increases, accuracy will also increase.
3. The algorithm requires less computing complexity compared with other published works.

The rest of the paper is organized as follows: In Section II, we provide a detailed description of the chessboard algorithm. In Section III, the simulation results from MATLAB and real-world data, comparisons with other papers are provided. In Section IV, the conclusion and future perspective are indicated.

II. ALGORITHM AND ANALYSIS

The initiation of the AMC algorithm typically begins with the constellation graph, which provides coordinates containing both real and imaginary parts of a signal, as exemplified by the Constellation Image (CI) method in [4]. As illustrated in Figure 1 after phase lock, high SNR signals can be manually distinguished by observing the distinct characteristics of various modulations, such as the locations where most points are concentrated. Signals are collected assuming the receiver is in symbol phase-lock, or the frequency error is sufficiently low so that the constellation rotation over the analysis period is less than a symbol boundary.

Because of the noise, constellation points are distributed around the actual constellation points according to the noise level. To facilitate a more straightforward yet effective distinction between different constellation points, we introduce the chessboard concept. As depicted in Figure 2, we first use small grids to divide the constellation plane and count the number of points located within each square, giving the constellation plane a chessboard-like appearance. Grid_scale is the side length figure of the squares used to divide the constellation graph as shown in Figure 2. Subsequently, the CAMC methodology quantifies the number of points in each square, providing insight into the points' distribution.

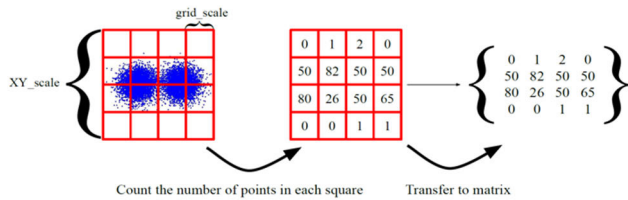


FIGURE 2. Concept of CAMC.

This information is then transformed into a matrix. Finally, by applying an inner product between the chessboard matrices of the symbols and the testing signal, we can maximize the similarity extent between the symbols and testing input modulation; larger results indicate a higher likelihood of the same modulation.

The detailed process of the CAMC algorithm is represented in Figure 3 and the equations will be analyzed below:

Chessboard Matrix Generation Process:

First, the initially set XY_scale (XY_S) and $grid_scale$ (GS) are used to generate a square matrix (M). The column/row size (S) of the matrix will be:

$$S = XY_S/GS \quad (1)$$

Then, I/Q pairs as input data will be inserted as I and Q respectively. Before passing to the next procedure, I and Q need to meet the if statement below:

$$\text{If}(|I| \leq XY_S \&\& |Q| \leq XY_S)$$

After the if statement above, I and Q will be transformed to coordinate data of the square matrix C_I and C_Q . C_I will be the row and C_Q will be the column. $R()$ refers to the round function.

$$C_I = R(I/GS) \quad (2)$$

$$C_Q = R(Q/GS) \quad (3)$$

At last, the square matrix M will be updated by the input:

$$M(C_I, C_Q) = M(C_I, C_Q) + 1 \quad (4)$$

The above process is for a single pair of I/Q . The total number of I/Q pairs could be decided by changing the number of symbols in the codes.

A. MATRIX INNER PRODUCTION PROCESS

After Matrix Generation Process, 8 matrices will be generated: 4 of which are from training data for BPSK, QPSK, 8PSK and 16QAM respectively and 4 input matrices with different angles of rotation. The equation of Inner production IP will be shown below:

$$IP(A, B) = \sum_{i=0, j=0}^S A(i, j) * B(i, j) \quad (5)$$

The inner production results will be gathered as below:

$$R = IP(B_M, I_M) \quad (6)$$

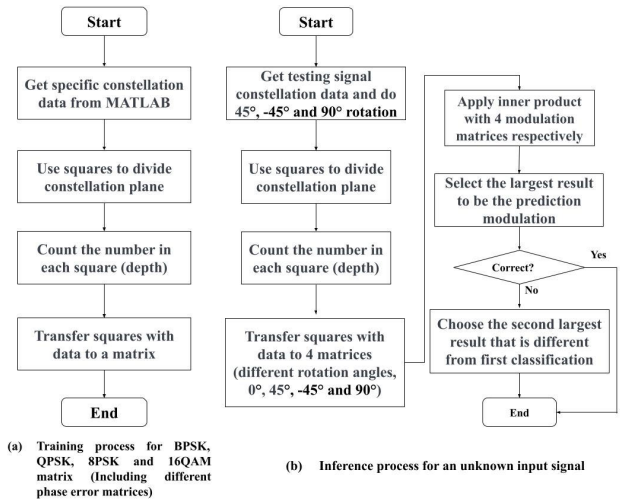


FIGURE 3. Flow Chart of CAMC Algorithm (a) Training (b) Inference.

Finally, the largest results among the 16 will be picked up as the final result (FR). And the modulation that refers to the FR will be the classification result.

$$FR = \max(\text{Among All } R) \quad (7)$$

To further improve the accuracy of this CAMC algorithm, a 2-pass classification method is applied. When the first classification is incorrect, the algorithm will pick up the second largest results which is not the same modulation type as the first classification.

The macro view of the CAMC process is shown below:

The chessboard process could be divided into 2 steps: “training” and “inference”. It will begin with the training symbols matrix generation, and then proceed to find the Inner product between the 4 input matrices (including rotation data) and 4 different modulation matrices. The modulation with the largest result will be the prediction.

B. TRAINING PROCESS

Before introducing the training process, it is essential to define the concept of depth. Depth refers to the number of points located in a single square on the chessboard, which will later be converted into an element in the chessboard matrix. First, users set a customized grid scale, which influences the accuracy and computational load of the results. The grid is then used to divide the constellation plane into squares. Subsequently, the algorithm counts the number of points located in the squares, representing the depths. The grid scale can be defined by users according to their needs; however, recommended numbers for hardware-friendly purposes include 0.1, 1/4, 1/8, and 1/16, among others. These numbers can be transferred into hardware using standard multipliers and shifters without imposing excessive precision requirements on the number system.

After setting the grid scale, we employ MATLAB to generate modulation data and obtain 4 classification matrices of BPSK, QPSK, 8PSK, and 16QAM, respectively. For each

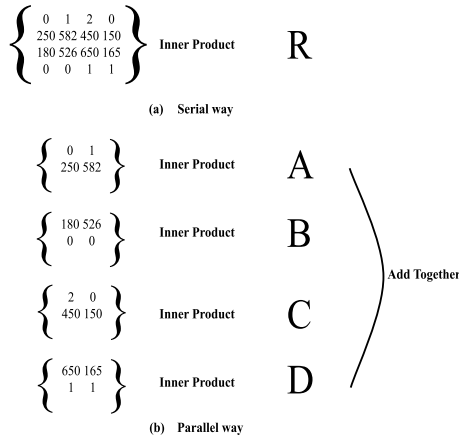


FIGURE 4. Parallel calculations in chessboard generation.

matrix, we generate a predetermined number of constellation points (defined by the users) with a fixed SNR. The process of creating these matrices serves as the training process for the algorithm, establishing a number in the classification matrices for the subsequent stage, referred to as the “inference” stage.

C. INFERENCE PROCESS

After completing the training process, we can utilize the generated matrices to classify a signal with unknown modulation.

After receiving input symbols, symbols will be rotated 45°, -45° and 90° respectively. Then these 4 sets of data (including non-rotation data) will proceed to generate 4 input matrices by using the same depth concept as in the training process.

Then, the inner productions between the input matrices and training matrices are calculated to maximize the difference. Each input matrix will proceed with inner production with training matrices of BPSK, QPSK, 8PSK, and 16QAM respectively and 16 results will be gathered in total. The modulation with the largest output is identified as the classification result. Furthermore, to improve the accuracy of CAMC, a 2-pass method is applied. If the first classification is wrong, then the algorithm will pick up the second-largest result whose modulation type is different from the first classification the result.

During this process, the number of input constellation points is also scalable to accommodate different demands. For higher accuracy, users can opt for a larger number of input data; for considerations of high speed and lower power-resource per classification, a smaller amount of data can be inserted.

D. PARALLELISM

This algorithm is well-suited for parallel implementation because there is no data dependency when performing all the inner products. Figure 4 and Figure 5 separately illustrate the parallel approach in chessboard generation and matrix inner product computation.

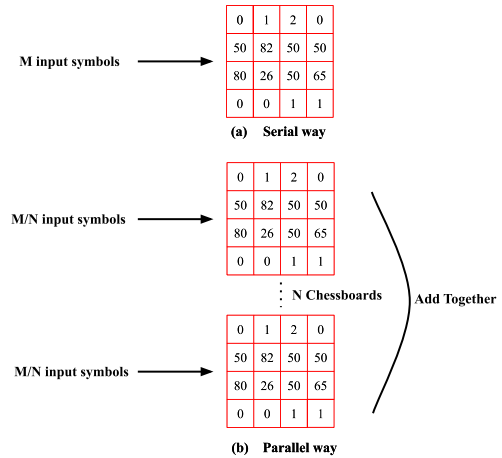


FIGURE 5. Parallel calculations in the matrix inner-product.

Figure 4 shows the generation N chessboards with the same size to accumulate the depth and add all depths in chessboards together after the process. In this way, we could shorten the processing time to 1/N based on the customized parallelism degree.

In Figure 5, an example inner product of the matrices is shown, it shows the inner product being found from whole matrices. However, as shown in Figure 5, the complete matrices can be decomposed into smaller ones to make the process parallel. After partial inner products, all results will be added together to get the classification. E.g., If a larger matrix is decomposed into N smaller partial matrices, then the total execution time can be reduced to 1/N of the original.

III. RESULTS AND COMPARISON

A. ACCURACY

MATLAB was employed to generate signals with BPSK, QPSK, 8PSK, and 16QAM modulation, respectively, using 1000 symbols under 20dB SNR and producing four classification matrices for each modulation. The accuracy was evaluated under various SNRs (from 0dB to 20dB), different grid scales, and varying numbers of input data. The outcomes were compared with those from other published works.

Table 1 presents the detection accuracy under different grid scales. Under the condition of 6000 inputs, grid scales of 0.1, 1/4, 1/8, and 1/16 were tested. The results indicate that accuracy increases under low SNR conditions of QPSK classification as the grid scale becomes smaller.

Table 2 shows the classification accuracy under varying numbers of input signals. The accuracy improves as the number of input data increases. As the input number increases, the results become more stable with higher accuracy. Meanwhile, for a large input number, only QPSK input will need a 2-pass method to classify while when a small number of inputs are applied, QPSK, 8PSK and 16QAM all need 2-pass method to classify. A larger number of input data provides the algorithm with more elements in the matrix, which amplifies the differences in the results and makes the modulation more distinguishable.

TABLE 1. Accuracies under different grid scales.

SNR(dB)		0	3	5	10	15	20
Grid Scale		Accuracy (%)					
Grid 0.25	BPSK	100	100	100	100	100	100
	QPSK	0	96*	100	100	100	100
(1/4)	8PSK	100	100	100	100	100	100
	16QAM	100	100	100	100	100	100
Grid 0.125 (1/8)	BPSK	100	100	100	100	100	100
	QPSK	0	96*	100	100	100	100
	8PSK	100	100	100	100	100	100
	16QAM	100	100	100	100	100	100
Grid 0.1	BPSK	100	100	100	100	100	100
	QPSK	0	99*	100	100	100	100
	8PSK	100	100	100	100	100	100
Grid 0.0625 (1/16)	BPSK	100	100	100	100	100	100
	QPSK	0	100*	100	100	100	100
	8PSK	99	100	100	100	100	100
	16CAM	100	100	100	100	100	100

TABLE 2. Accuracy under different numbers of input symbols.

		0dB (%)	3dB (%)	5dB (%)	10dB (%)	15dB (%)	20dB (%)
symbol 2000	BPSK	100	100	100	100	100	100
	QPSK	0	97*	100	100	100	100
	8PSK	100*	100*	100	100	100	100
	16QAM	98	100	100	100	100	100
symbol 4000	BPSK	100	100	100	100	100	100
	QPSK	0	89*	100	100	100	100
	8PSK	100	100	100	100	100	100
	16QAM	100	100	100	100	100	100
symbol 5000	BPSK	100	100	100	100	100	100
	QPSK	0	100	100	100	100	100
	8PSK	100	100	100	100	100	100
	16QAM	100	100	100	100	100	100
symbol 6000	BPSK	100	100	100	100	100	100
	QPSK	0	99	100	100	100	100
	8PSK	100	100	100	100	100	100
	16QAM	100	100	100	100	100	100

* : 2 Pass result. Not stable, pick up the most frequent results.

Figure 6 and Table 3 present comparisons between this work and other published works [2], [4], and [9]. In Figure 6, the worst cases are highlighted, while in Table 3, the accuracy rate for different approaches from the papers is provided (NA denotes data not mentioned). The results of 6000 input symbols and a grid scale of 0.1 were used for comparison. Under high SNR conditions, CAMC performs exceptionally well, with an accuracy of 100% under SNR conditions above 5dB.

At 0dB, only QPSK presents a challenge, while other modulation detections achieve over 100% accuracy. QPSK classification under a low SNR poses a difficult classification in other papers because of the similar characteristics between QPSK and 8PSK. For example, the accuracy of QPSK classification could not exceed 60% in [4] with the results of 54% (CI [4]), 57% (GRF [4]) respectively, and lower than 70% in [2] (69.3% using fixed threshold method).

This issue becomes worse in the CAMC algorithm as the constellation graphs of QPSK and 8PSK are similar as shown in Figure 7.

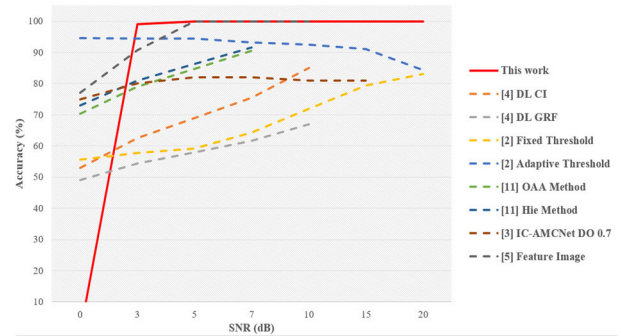


FIGURE 6. Comparisons between this work and other works.

TABLE 3. Comparisons between this work and other works.

SNR (dB)	0	3	5	7	10	15	20
	Accuracy (%)						
Our work	0	99	100	100	100	100	100
[4] DL CI	53	NA	NA	NA	85	NA	NA
[4] DL GRF	49	NA	NA	NA	67	NA	NA
[2] Fix Threshold	56	NA	59	NA	72	79	83
[2] Adapt Threshold	95	NA	95	NA	92	91	84
[11] OAA	70	NA	NA	91	NA	NA	NA
[11] Hie	73	NA	NA	92	NA	NA	NA
[3] IC-AMCNet DO 0.7	75	80	82	82	81	81	NA
[5] Feature Image	77	NA	100	NA	100	NA	NA

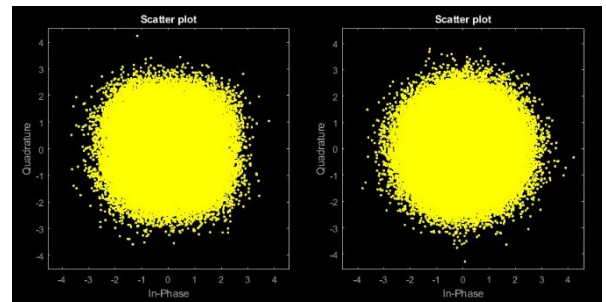


FIGURE 7. QPSK and 8PSK constellation graph under 0dB.

Also, after grid_scale division, the QPSK, 8PSK, QPSK_8PSK common elements, and QPSK_8PSK non-common elements are shown below in Figure 8. From the figure, we could see that most of the elements in the 2 modulations have the same positions. The same positions elements are counted to 1631 whereas 8PSK has 2022 elements and QPSK has 1942 elements in total.

In the CAMC algorithm, an inner production will be processed. In the inner product, when an element from any matrix is 0, then this pair of elements could be ignored in the calculation since it gives 0. Therefore, in the CAMC algorithm, the inner production could be simplified into the “same position production with training matrices” as in Figure 8. In Figure 9, red centers are from the training matrix of QPSK while red and blue centers are from the 8PSK matrix. 4 centers are the same between these two matrices. Yellow points indicate 0dB

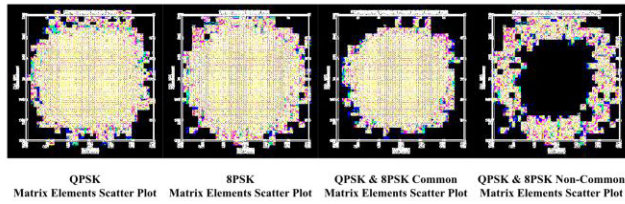


FIGURE 8. QPSK, 8PSK, QPSK_8PSK common elements, and QPSK_8PSK non-common elements.

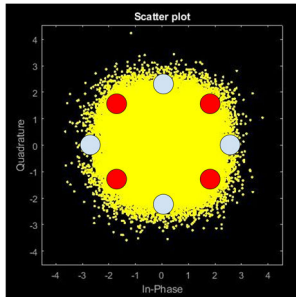


FIGURE 9. Inner production simplification in the CAMC.

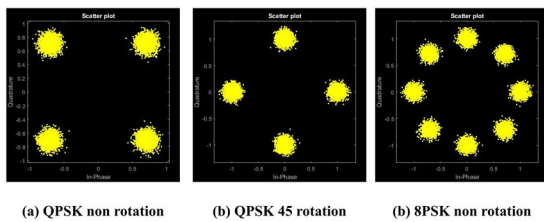


FIGURE 10. 45° rotation QPSK and 8PSK constellation graphs.

QPSK input constellation graph. Because 6000 symbols are inserted, all 8 centers will have a similar number of symbols. Besides, as the rotation input matrix is utilized when the rotation angle is 45° of QPSK, there will not be any common center with the QPSK training matrix. However, there will be 8PSK common centers instead, as shown in Figure 10. Therefore, after inner production, the result of the 8PSK will be much larger than the QPSK result. It leads to the condition that all the QPSK signals at 0dB will be classified as 8PSK. But this could also be optimized. Since all the QPSK will be classified to 8PSK at 0dB, therefore the result will be a choice only between 8PSK and QPSK. However, when the SNR is 3dB, the signal power versus noise ratio is around 2. Under this condition, the difference between QPSK and 8PSK will be enhanced a lot, which leads to a large gap between accuracies under 0dB and 3dB.

The confusion matrix of this CAMC design (6000 input symbols, grid scale = 0.1) is shown below with SNR of 3dB, 5dB, 10dB, and 20 dB respectively.

Confusion matrix of recognition at 3dB SNR (%)

BPSK	100	0	0	0
QPSK	0	99	1	0
8PSK	0	0	100	0
16QAM	0	0	0	100
	BPSK	QPSK	8PSK	16QAM

Confusion matrix of recognition at 5dB SNR (%)

BPSK	100	0	0	0
QPSK	0	100	0	0
8PSK	0	0	100	0
16QAM	0	0	0	100
	BPSK	QPSK	8PSK	16QAM

Confusion matrix of recognition at 10dB SNR (%)

BPSK	100	0	0	0
QPSK	0	100	0	0
8PSK	0	0	100	0
16QAM	0	0	0	100
	BPSK	QPSK	8PSK	16QAM

Confusion matrix of recognition at 20dB SNR (%)

BPSK	100	0	0	0
QPSK	0	100	0	0
8PSK	0	0	100	0
16QAM	0	0	0	100
	BPSK	QPSK	8PSK	16QAM

Except for errors caused by noise, synchronization error is another factor that could influence the signals. It is caused by delays in the circuits [16]. However, other publications of AMC applications did not mention and test for synchronization errors in recent years, therefore, it is fair that this error test is not shown in this paper.

Moreover, some lab-gathered (real-world) data are tested to be classified by the CAMC algorithm. The signals were generated at mmwave frequencies using a Rhode & Schwarz SMM100A. They were then transmitted over a cable to be received by a Keysight PXA N9030B signal analyzer. Below are the confusion matrices. These real-world lab-gathered data contain synchronization errors. There are 20000 symbols for each modulation and are divided into 10 groups (each group contains 2000 symbols). These groups are inserted into the CAMC algorithm. The results show that by using the CAMC algorithm, lab-gathered data could be 100% correctly classified.

Confusion matrix of lab data (real-world) (%)

BPSK	10	0	0	0
QPSK	0	10	0	0
8PSK	0	0	10	0
16QAM	0	0	0	10
	BPSK	QPSK	8PSK	16QAM

B. COMPUTATION DEMANDS AND HARDWARE FRIENDLY

When implementing our algorithm into hardware with time and power constraints, it is sometimes necessary to adjust the algorithm, such as modifying precisions or divisions, which may result in a sacrifice of precision. However, our algorithm is hardware-friendly and does not require any modifications. Additionally, our algorithm is computationally simple and can meet future challenges of the short time window for classifications.

The number of operations is counted below in Table 4 Table 5 and Table 6.

In (1), SN is the number of input symbols, XY_scale represents the larger absolute value of both the real and imaginary

TABLE 4. Number of operations in input data rotation.

Operations	Number
sqrt	SN
Power of 2	2*SN
&	2*SN
OR	SN
atan	SN
/	SN
cos	SN
Sin	SN
*	2*SN

TABLE 5. Number of operations in one matrix generation.

Operations	Number
+	3*SN
Comparison	4*SN
*	2*SN
&	3*SN

TABLE 6. 4 Matrix inner production operations.

Operations	Number
* or bit shifting	$8 * \left(\frac{2 * XY_{scale}}{grid_{scale}} \right)^2$
+	$4 * \left(\frac{2 * XY_{scale}}{grid_{scale}} \right)^2$

parts of the data). Grid_scale refers to the user-determined grid scale, while symbol_no indicates the number of input signal data chosen by the user. The second row demonstrates a choice between shift or multiplication. If users select a grid scale of 0.1, then multiplication is used; otherwise, bit-shifting is employed.

Compared to other related work that utilizes deep learning or AI-assisted methods, this approach demands less computing power and holds the potential for high performance-power efficiency implementation. Furthermore, this methodology is well-suited for FPGA applications, as it requires much less and simpler operations. Additionally, users can choose grid scales of 0.1, 1/4, 1/8, and 1/16, which can be converted to bit-shifting and multiplication instead of performing divisions.

IV. CONCLUSION AND FUTURE WORK

In this paper, we propose a Chessboard-based Automatic Modulation Classification (CAMC) algorithm that employs a depth concept to automatically classify four modulations: BPSK, QPSK, 8PSK, and 16QAM. This algorithm offers scalable variables to accommodate different user requirements. The CAMC algorithm is hardware-friendly and can be easily implemented on FPGA for FPGA-based SDR applications.

From the test results, this algorithm demonstrates higher performance compared to other works, particularly in high SNR conditions.

Meanwhile, for real-world lab-gathered data, this CAMC algorithm still provides a 100% accurate result.

The algorithm is also scalable, allowing users to adjust the accuracy by varying the number of input symbols and using different grid scales for improved classification under low SNRs.

Compared to other published papers, especially those employing AI or deep-learning-assisted methods, the proposed algorithm demands significantly less computing power and holds the potential for low-power implementations. Additionally, the CAMC algorithm's simplicity substantially shortens the design period when transferring it to hardware, such as FPGAs.

However, there are still some problems that need to be tackled. The QPSK under 0dB will give 0% accuracy, which is even below the random guess of 25%. In future work, the solution of this scenario will be developed, and this algorithm will be implemented on an FPGA and tested using live data in the future. Subsequently, it will be integrated into an SDR to evaluate its functionality and potential in a real-world scenario.

REFERENCES

- [1] Y. Motwani, P. Saraswat, S. Aggarwal, S. Aniket, R. M. Awari, and A. Bagubali, "Analysis of various neural network architectures for automatic modulation techniques," in *Proc. Innov. Power Adv. Comput. Technol. (i-PACT)*, Kuala Lumpur, Malaysia, Nov. 2021, pp. 1–7, doi: 10.1109/i-PACT52855.2021.9696629.
- [2] S. Ghodeswar and P. G. Poonacha, "An SNR estimation based adaptive hierarchical modulation classification method to recognize M-ary QAM and M-ary PSK signals," in *Proc. 3rd Int. Conf. Signal Process., Commun. Netw. (ICSCN)*, Chennai, India, Mar. 2015, pp. 1–6, doi: 10.1109/ICSCN.2015.7219867.
- [3] A. P. Hermawan, R. R. Ginanjar, D.-S. Kim, and J.-M. Lee, "CNN-based automatic modulation classification for beyond 5G communications," *IEEE Commun. Lett.*, vol. 24, no. 5, pp. 1038–1041, May 2020, doi: 10.1109/LCOMM.2020.2970922.
- [4] Y. Sun and E. A. Ball, "Automatic modulation classification using techniques from image classification," *IET Commun.*, vol. 16, no. 11, pp. 1303–1314, Jul. 2022, doi: 10.1049/cmu2.12335.
- [5] J. H. Lee, K.-Y. Kim, and Y. Shin, "Feature image-based automatic modulation classification method using CNN algorithm," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIC)*, Okinawa, Japan, Feb. 2019, pp. 1–4, doi: 10.1109/ICAIC.2019.8669002.
- [6] N. Kassri, A. Ennouary, S. Bah, and H. Baghdadi, "A review on SDR, spectrum sensing, and CR-based IoT in cognitive radio networks," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 6, pp. 100–121, 2021, doi: 10.14569/IJACSA.2021.0120613.
- [7] Y. H. Lin, Q. Wang, J. S. Wang, L. Shao, and J. Tang, "Wireless IoT platform based on SDR technology," in *Proc. IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things IEEE Cyber. Phys. Social Comput.*, Aug. 2013, pp. 2245–2246, doi: 10.1109/GreenCom-iThings-CPSCom.2013.426.
- [8] M. M. Roganovic, A. M. Neskovic, and N. J. Neskovic, "Application of artificial neural networks in classification of digital modulations for software defined radio," in *Proc. IEEE EUROCON*, Saint Petersburg, Russia, May 2009, pp. 1700–1706, doi: 10.1109/EURCON.2009.5167872.
- [9] X. Wei, X. Qi, L. Xiao, Z. Shi, and L. Huang, "Software-defined radio based on Cortex-A9," in *Proc. 7th Int. Conf. Commun. Netw. China*, Kunming, China, Aug. 2012, pp. 758–761, doi: 10.1109/China-Com.2012.6417585.
- [10] A. Karlsson, J. Sohl, J. Wang, and D. Liu, "EPUMA: A unique memory access based parallel DSP processor for SDR and CR," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Austin, TX, USA, Dec. 2013, pp. 1234–1237, doi: 10.1109/GlobalSIP.2013.6737131.
- [11] X. Teng, P. Tian, and H. Yu, "Modulation classification based on spectral correlation and SVM," in *Proc. 4th Int. Conf. Wireless Commun., Netw. Mobile Comput.*, Dalian, China, Oct. 2008, pp. 1–4, doi: 10.1109/WiCom.2008.409.

- [12] Y. Hu, Y. Liu, and Z. Liu, "A survey on convolutional neural network accelerators: GPU, FPGA and ASIC," in *Proc. 14th Int. Conf. Comput. Res. Develop. (ICCRD)*, Shenzhen, China, Jan. 2022, pp. 100–107, doi: [10.1109/ICCRD54409.2022.9730377](https://doi.org/10.1109/ICCRD54409.2022.9730377).
- [13] P. K. Meher, "An optimized lookup-table for the evaluation of sigmoid function for artificial neural networks," in *Proc. 18th IEEE/FIP Int. Conf. VLSI Syst.-on-Chip*, Madrid, Spain, Sep. 2010, pp. 91–95, doi: [10.1109/VLSISOC.2010.5642617](https://doi.org/10.1109/VLSISOC.2010.5642617).
- [14] P. S. Nair, S. Koppa, and E. B. John, "A comparative analysis of coarse-grain and fine-grain power gating for FPGA lookup tables," in *Proc. 52nd IEEE Int. Midwest Symp. Circuits Syst.*, Cancun, Mexico, Aug. 2009, pp. 507–510, doi: [10.1109/MWSCAS.2009.5236045](https://doi.org/10.1109/MWSCAS.2009.5236045).
- [15] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5G networks for the Internet of Things: Communication technologies and challenges," *IEEE Access*, vol. 6, pp. 3619–3647, 2018, doi: [10.1109/ACCESS.2017.2779844](https://doi.org/10.1109/ACCESS.2017.2779844).
- [16] A. A. Nasir, S. Durrani, H. Mehrpouyan, S. D. Blostein, and R. A. Kennedy, "Timing and carrier synchronization in wireless communication systems: A survey and classification of research in the last 5 years," *EURASIP J. Wireless Commun. Netw.*, vol. 2016, no. 1, Dec. 2016, Art. no. 180, doi: [10.1186/s13638-016-0670-9](https://doi.org/10.1186/s13638-016-0670-9).
- [17] Y. Linn, "New structures for modulation classification and SNR estimation with applications to cognitive radio and software defined radio," in *Proc. IEEE 27th Can. Conf. Electr. Comput. Eng. (CCECE)*, Toronto, ON, Canada, May 2014, pp. 1–8, doi: [10.1109/CCECE.2014.6900915](https://doi.org/10.1109/CCECE.2014.6900915).



YUQIN ZHAO received the B.E. degree in electrical and computer engineering from Tamkang University, New Taipei City, Taiwan, in 2021, and the M.Sc. degree in electronics from Nanyang Technological University, Singapore, in 2022. He is currently pursuing the Ph.D. degree in electronic and electrical engineering with The University of Sheffield, Sheffield, U.K. His research interests include the development of an FPGA accelerator for AI and SDR, the SDR implementation of SDR application, and high-level synthesis environment for SDR and AI applications on FPGAs.



WILLIAM C. J. GAVIN received the M.Eng. degree (Hons.) from The University of Sheffield, in 2020, where he is currently pursuing the Ph.D. degree in machine learning applied to the physical layer of communications systems. His research interests include the application of machine learning algorithms to communication systems and hardware acceleration for artificial intelligence.



TIANTAI DENG (Member, IEEE) received the B.Eng. degree from the Harbin Institute of Technology, in 2015, and the Ph.D. degree from Queen's University Belfast, in 2019. He is currently a Lecturer with The University of Sheffield. Prior to his career as an academic, he was a Senior Engineer with HiSilicon, Huawei. His research interests include hardware acceleration for image processing, deep learning, and high-level design environments.



EDWARD A. BALL (Member, IEEE) was born in Blackpool, U.K., in November 1973. He received the Master of Engineering degree (Hons.) in electronic systems engineering from the University of York, York, U.K., in 1996. After graduating, he worked in the industry for 20 years, first spending 15 years working as an Engineer, a Senior RF Engineer, and finally as a Principal RF Engineer with Cambridge Consultants Ltd., Cambridge, U.K. Then, he spent five years as a Principal RF Engineer and a Radio Systems Architect with Tunstall Healthcare Ltd., Whitley, U.K. In November 2015, he joined the Department of Electronic and Electrical Engineering, The University of Sheffield, Sheffield, U.K., where he is currently a Reader in RF engineering. His research interests include radio technology, from RF system design, RF circuit design (sub-GHz to mm-wave), and the application of radio technology to real-world industrial and commercial problems. He has a particular passion for RF hardware design. He is a member of the IET and he is also a Chartered Engineer.



LUKE SEED is currently a Senior Lecturer with the Department of Electronic and Electrical Engineering, The University of Sheffield. His research interests extend from chip design through to holographic lithography.

• • •