

# Implementation of systolic array based SVM classifier using multiplierless kernel

BHASWATI MANDAL, MANASH P. SARMA and KANDARPA KUMAR SARMA

Gauhati University, Deptt. of Electronics and

Communication Engineering, Guwahati-781014, Assam, INDIA.

bhaswatimndl@gmail.com, manashpelsc@gmail.com, kandarpaks@gmail.com

Nikos Mastorakis

Techn.Univ Sofia

Technical University - Sofia

Sofia 1000, "Kl. Ohridski" 8, Bulgaria

mastor@tu-sofia.bg

*Abstract:* In the field of image processing, computer vision, bio-informatics, classification of data is an area of study for many researchers and scholars from the beginning. SVMs are dynamic and powerful learning methods which provide excellent generalization performance for a wide range of regression and classification problems. Previous software implementations of SVMs have reported high classification accuracy. But software designs can not meet the real time requirements because these designs can not take the advantage of parallelism inherent in the SVM algorithm. Thus, the hardware implementation of SVM can efficiently decrease total simulation time and synthesis time. In the field of classification efficient utilization of power is one of the major concern. So, in this paper we mainly concern about decreasing hardware complexity and reduction of power of SVM design. For reduction of power we are proposing here the multiplierless kernel for classification in hardware, instead of using conventional vector product kernel. We have trained SVM using binary linearly, nonlinearly separable and multiclass data in Matlab. Then the extracted trained parameters are used in the hardware for classification purpose. For data flow from processing element (PE) to PE in hardware, we are using parallelly pipelined systolic array system.

*Key-Words:* Support Vector machine, Multiplierless kernel operation, Systolic array.

4.06.14

## 1 Introduction

In classification problems, SVM classifiers often show superior recognition efficiency than the other classifiers. So, it becomes quite popular after its introduction by Cortes and Vapnik in 1995 [1]. For its high classification accuracy, the embedded designing of SVM for image processing, bio-informatics, object detection has been growing interest. SVMs work on the principle of decision boundary. The boundary that separates two discriminant classes with high accuracy. SVMs are basically binary classifiers. To create multiclass classification, we are mainly using here one against all algorithm for which all the patterns from class  $x$  are trained as positive instances and the patterns from all other classes are trained as negative instances for classifier  $x$  [1], [2]. The class which gives highest output is the class of the new test sample. In

the field of classification efficient utilization of power is one of the major concern. So, in this paper we mainly concern about decreasing hardware complexity and reduction of power of SVM design. For reduction of power we are proposing here the multiplierless kernel for classification in hardware instead of using conventional vector product kernel. We have trained SVM using binary linearly, nonlinearly separable and multiclass data in Matlab. Then the extracted trained parameters are then used in the hardware for classification purpose. For data flow from processing element (PE) to PE in hardware, we are using parallelly pipelined systolic array system.

The rest of the sections of this paper are organized as follows. Section 2 provides an explanation on the working principle of SVM and a brief discussion on previous related works. The proposed architecture is

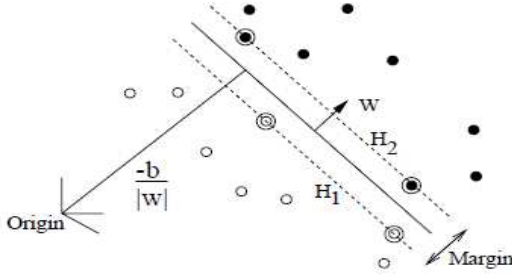


Figure 1: Geometrical representation of the SVM margin

described in Section 3. The experimental details and results are discussed in Section 4. Lastly, Section 5 provides conclusion and future work.

## 2 SVM Background and Related work

Here, working principle of SVM and the works related to implementation of SVM are discussed.

### 2.1 SVM Background

SVM is based on the concept of decision planes that separates two classes of data. The data set containing samples are given as input to the SVM, which construct the separating hyperplane that separates two classes of data. The hyperplane that best separates data of two classes are called maximum margin hyperplane and the samples that resides nearest to the hyperplane are called Support vectors (SV) [2]. These SVs are then used for classifying new data. Geometrical representation of the SVM margin is shown in Fig 1. Let the training data labeled as  $\{X_i, y_i\}$ ,  $i = 1, 2, \dots, l$ ,  $y_i \in \{-1, 1\}$ ,  $X_i \in R^d$ . The hyperplane is the plane that separates positive samples from the negative samples. The point  $X$  that lies on the hyperplane satisfies  $wx + b = 0$ , where  $w$  is normal to the hyperplane,  $\|b\|/\|w\|$  is the perpendicular distance from the hyperplane to the origin and  $\|w\|$  is the Euclidian distance. Let  $d_+$  be the shortest distance of the hyperplane from the positive side and  $d_-$  from the negative side respectively. Then the equation for margin of the separating hyperplane becomes  $d_+ + d_-$ . Let all the training samples satisfies the following conditions:

$$x_i \cdot w + b \geq +1 \text{ for } y_i = +1 \quad (1)$$

$$x_i \cdot w + b \leq -1 \text{ for } y_i = -1 \quad (2)$$

These two equations can be combined as following:

$$y_i \cdot (x_i \cdot w + b) - 1 \geq 0, \forall i \quad (3)$$

When samples from two classes are not linearly separable then it is projected to higher dimension data using kernel trick. The most innovative kernel designs which are widely used because of their efficiency in mapping data to higher dimensional space are listed below [2]:

1. Linear:  $K(\vec{x}, \vec{z}) = (\vec{x} \cdot \vec{z})$
2. Polynomial:  $K(\vec{x}, \vec{z}) = (1 + (\vec{x} \cdot \vec{z}))^d$
3. Sigmoid:  $K(\vec{x}, \vec{z}) = \tan((\vec{x} \cdot \vec{z}) + \theta)$
4. Radial Basis function:  $K(\vec{x}, \vec{z}) = \exp((x - z)^2 / (2\sigma^2))$

The main decision function of SVM classification is given by equation (4).

$$z_j = \text{sign}(\sum a_i y_i k(x_i, s_i) + b) \quad (4)$$

### 2.2 Related Work

During the last few years, hardware designing of SVM has recieved lots of interest. Many designs have been developed in this field. Yet there are lot more refinement to be done. In [3], the authors utilize the uniformity behavior of the SVM decision function in the integrated vision system. The main module used in this system is SVM classifier. The authors proposed a parallel implementation on an FPGA programmed with VHDL for the reduction of the computation time. In [4], the authors proposed an SVM learning algorithm and elaborate its implementation on a field programmable gate array (FPGA). In [5], the authors present a parallel architecture for SVM to be implemented on Xilinx FPGA. Here they used thousands of complex classification patterns from the high energy physics to obtain the results and also compared the performance of the architecture with the simpler sequential architecture. In [6], the authors present a massively parallel FPGA based coprocessor. To take the advantage of large amount of parallelism in data of this application, both SVM training and classification is implemented in this coprocessor. In [7], the authors introduce a design called Systolic Chain of Processing Elements (SCOPE) which was the first attempt of realization of generic systolic array in SVM for object detection and and describes its embedded audio and video application. This design provides efficient memory management, reduced complexity and efficient data transfer mechanisms. As the size of the chain and kernel module can be changed in plug, the

proposed architecture is generic and scalable and any changes can be done without effecting the overall architecture. In [8], the authors designed a high performance circuit which supports both linear and nonlinear classification. Concerning the efficiency of classification a  $48 \times 96$  or  $64 \times 64$  sliding window with window strides is used. The circuit size is minimized here by sharing most of the resources used for linear and nonlinear classification. In [9], a power aware hardware implementation of multiclass SVM on FPGA using systolic array architecture is presented. Here, the authors used reconfiguration method for power reduction and compared it to the same design before reconfiguration.

### 3 Proposed Approach

The proposed block diagram of our design is shown in Figure 2. We have trained the SVM using two groups of linearly separable setosa and non-setosa data of Fisheriris, two groups of nonlinearly separable data and 3 class data in Matlab separately. These generates SVs, alpha values and class labels separately for all the 3 problems. These values are then used for the classification purpose. As power reduction becomes a crucial factor now a days. We are proposing the use of multiplierless kernel instead of vector product kernel. The working of our entire design is summarized below:

- Random test samples are given input to each PE.
- Then the multiplierless kernel operation is performed in each PE. This yields a scalar value which is then multiplied with alpha labels and class labels. The value of class label can be +1 or -1.
- All these operations are parallaly performed in each PE.
- The outputs from each PE are then stored in registers and these values are added classwise to their corresponding bias values.
- The class with maximum value is the class of the test sample.

From the training phase of the above 3 classification problems we got 8 SVs and alpha values for binary linear case, 24 SVs and alpha values for binary non linear case and 74 SVs and alpha values for multiclass problem. The dimensions of different matrices of binary linear classification, binary nonlinear classification and multiclass classification is given in Table 1, Table 2 and Table 3 respectively.

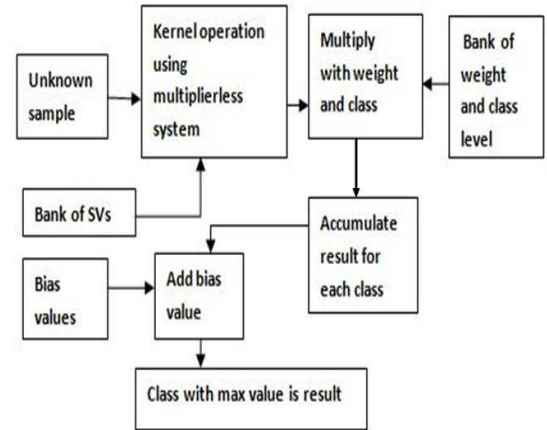


Figure 2: The Proposed architecture of SVM

Table 1: Dimension for different matrices for linear classification

Class	Test Vector	SV	Kernel	decision fn
setosa	$1 \times 2$	$2 \times 3$	$1 \times 3$	$1 \times 1$
non setosa	$1 \times 2$	$2 \times 5$	$1 \times 5$	$1 \times 1$

Table 2: Dimension for different matrices for nonlinear classification

Class	Test Vector	SV	Kernel	Decision fn
class 0	$1 \times 2$	$2 \times 12$	$1 \times 12$	$1 \times 1$
class 1	$1 \times 2$	$2 \times 12$	$1 \times 12$	$1 \times 1$

Table 3: Dimension for different matrices for multi-class classification

Class	Test Vector	SV	Kernel	Decision fn
class 1	$1 \times 2$	$2 \times 12$	$1 \times 12$	$1 \times 1$
class 2	$1 \times 2$	$2 \times 30$	$1 \times 30$	$1 \times 1$
class 3	$1 \times 2$	$2 \times 32$	$1 \times 32$	$1 \times 1$

In the multiplierless block we are using Canonic Signed Digit (CSD) to reduce the maximum number of adders. Simplifying this kernel block results in reduction of power used in multiplierless block. Again, we are using Systolic Array for speeding up the entire system. We here provide a brief description of Systolic Array and CSD algorithms.

#### 3.1 SA-Basic architecture

Systolic architecture represent a network of PE that rhythmically compute and pass data through the system. These PE regularly pump data in and out such that a regular flow of data is maintained. As a result, systolic system features modularity and regular-

ity, which are important properties for VLSI design [11]. It was invented by Kung and Leiserson (1978). Ever since Kung proposed the systolic model its elegant solutions to demanding problems and its potential performance have attracted great attention [12].

### 3.2 CSD Representation

CSD is a number system by which we can represent a floating point number in two's complement form. The representation uses -1, 0, +1 (or -, 0, +) symbols only. With each position denoting the addition and subtraction of power of 2 [12]. These encoding techniques contains 33 % fewer non-zero elements than 2's complement form which leads to efficient implementations of add/subtract networks in hardwired digital signal processing (DSP). The properties of a CSD number are listed below:

- No two consecutive bits in a CSD number are non-zero.
- CSD representation of a particular number is always unique.
- A CSD representation contains minimum possible number of ones.

The general representation of a 2's complement form is  $\sum a_i 2^i$ , where  $a_{n-1} \in \{-1, 0, 1\}$ ,  $a_i \in \{0, 1\}$ ,  $i = 0, K, n - 2$  and  $n$  is the word length. If we insert -1 in the value set of bit representation, it becomes  $a_i \in \{-1, 0, 1\}$ ,  $i = 0, K, n - 1$ , and  $a_j \cdot a_{j+1} = 0$ ,  $j = 0, \wedge, n - 2$ . The constant  $a_i$  is said to be in CSD representation. CSD algorithm is applied to the input test vectors, then conventional shift and add operations are applied [13], [14]. This algorithm is efficiently applied in FIR filter design to reduce the complexity of hardware. Here also we are applying this algorithm in SVM with the same motive.

## 4 Results and Discussion

The output of CSD conversion block is shown in Table 4. The CSD digit is binary encoded into a sign bit *xis* and a magnitude bit *xim*. Under the sign-magnitude encoding 0 = 00; 1 = 01; -1 = 11.

### 4.1 Binary linear classifier hardware implementation

We have designed the binary linear classifier in XILINX 7vx485tffg1157-2. The primitive and black box

Table 4: Values after applying CSD Algorithm in hardware

Test Vector	Binary	CSD digit (In Xilinx)	
4.2	100.0011001100	xis	0000000010001
		xim	0010001010101
4.5	100.1	xis	0000000000000
		xim	0010010000000
5	101	xis	0000000000000
		xim	0010100000000
5.5	101.1	xis	0010100000000
		xim	0101010000000
6	0110	xis	0001000000000
		xim	0101000000000
3	0011	xis	0000100000000
		xim	0010100000000

Table 5: Primitive and Black Box usage for multiplierless kernel of binary linear SVM

GND	1	MUXCY	789
INV	372	MUXF7	6
LUT1	26	VCC	1
LUT2	187	XORCY	457
LUT3	294	IBUF	26
LUT4	217	OBUF	37
LUT5	232	IO buffers	63
LUT6	682	DSP48E1	0

Table 6: Device utilization summary for multiplierless kernel of binary linear SVM

Device: xc5vlx110t	Utilized	Available	% Utilization
Number of Slice LUTs:	2010	303600	0.7%
Number used as Logic:	2010	303600	0.7%
Number with an unused Flip Flop:	2010	2010	100%
Number with an unused LUT	0	2010	0%
Number of bonded IOBs:	63	600	10%
Number of DSP48E1	0	2800	0%

uses of LUTs, DSP4s and clocks of binary linear classifier using multiplierless kernel is given in Table 5. The design utilization summary using multiplierless kernel is given in Table 6. This is obtained from the synthesis report of the design in XILINX. The power is measured using XILINX POWER ESTIMATOR (XPE) 14.1. This version is used for power measurement of 7 series FPGA. The on chip power summary report of both the vector product kernel module and multiplierless kernel for the binary linear classification is given in Table 7.

From Table 6, we can interpret that LUT usage

Table 7: On- Chip Power Summary for binary linear SVM

On- chip	vector product(W)	Multiplierless(W)
Clock	0.005	0.002
Logic:	0.012	0.014
PLL	0.114	0.114
Others	0.332	0.332
BRAM	0.000	0.000
I/Os	0.766	0.766
DSPs	0.012	0
Device Statistics	0.251	0.251
<b>Total</b>	<b>1.493</b>	<b>1.479</b>

in multiplierless kernel is 0.7% but the dsp usage is 0%. We can justify the usage of large number of LUTs. As we are using CSD algorithm and add and shift operation in multiplierless binary SVM instead of vector multiplication, the logic increases in multiplierless binary SVM. But this has no effect on the power factor which is demonstrated in Table 7. Here as the resource utilization (LUT) is higher in multiplierless kernel so the logic consumes higher power. All the other power consumption factors like clock, PLL, I/O, device statistics and others consume almost same power for multiplierless kernel and the vector product kernel, only we can find a difference in power consumption in case of the dsp. As dsp consumes more power in vector product kernel, the total power consumption decreases from 1.493 to 1.479 using multiplierless kernel. So, we have achieved around 1% reduction in power in binary linear SVM.

#### 4.2 Binary non linear classifier hardware implementation

This design is implemented in XILINX 7vx485tffg1157-2. The primitive and black box uses of LUTs, DSP4s and clocks for binary non linear classifier using multiplierless kernel is given in Table 8. The design utilization summary using multiplierless kernel is given in Table 9. The synthesis report estimates the resources used for every design implemented in Xilinx. The power is measured using XILINX POWER ESTIMATOR (XPE) 14.1. This version is used specifically for the 7 series FPGA. The on chip power summary report of both the vector product kernel module and multiplierless kernel for the binary non linear classification is given in Table 10.

In the nonlinear case, for multiplierless kernel also 515 numbers of dtps are used. This is because we are using polynomial kernel of order 4. So, DSPs are extracted for the exponent operation of polynomial kernel. On chip power summary for

Table 8: Primitive and Black Box usage for multiplierless kernel of binary non linear SVM

GND	1	MUXCY	2225
INV	2087	MUXF7	3
LUT1	1760	VCC	1
LUT2	3534	XORCY	5436
LUT3	4030	IBUF	26
LUT4	6379	OBUF	1
LUT5	1177	IO buffers	27
LUT6	46	DSP48E1	515

Table 9: Device utilization summary for multiplierless kernel of binary non linear SVM

Device: XILINX xc7vx485t	Utilized	Available	% Utilization
No. of Slice LUTs:	19023	303600	7%
No. used as Logic:	19023	303600	7%
No. with an unused Flip Flop:	19023	19023	100%
No. with an unused LUT	0	19023	0%
No. of bonded IOBs:	27	600	4%
No. of DSP48E1	515	2800	18%

Table 10: On- Chip Power Summary for binary non linear SVM

On- chip	vector product(W)	Multiplierless(W)
Clock	0.016	0.014
Logic:	0.089	0.134
PLL	0.114	0.114
Others	0.332	0.332
BRAM	0.000	0.000
I/Os	0.766	0.766
DSPs	0.478	0.378
Device Statistics	0.255	0.255
<b>Total</b>	<b>2.051</b>	<b>1.994</b>

binary nonlinear SVM is demonstrated in Table 10. Here as the resource utilization (LUT) is higher in multiplierless kernel so the logic consumes higher power in multiplierless kernel than the vector product kernel. All the other power consumption factors like clock, PLL, I/O, device statistics and others consumes almost same power for multiplierless kernel and the vector product kernel except the dsp. But as the power consumption due to dsp is more in vector product kernel, the total power consumption decreases from 2.051 to 1.994 due to the use of multiplierless kernel. So we are successful in around 2.7% reduction in power in case of binary non linear SVM.

Table 11: Primitive and Black Box usage for multiplierless kernel of multiclass SVM

GND	211	MUXCY	3352
INV	2695	MUXF7	0
LUT1	154	VCC	41
LUT2	4913	XORCY	3985
LUT3	7612	IBUF	26
LUT4	12874	OBUF	2
LUT5	712	IO buffers	28
LUT6	4400	DSP48E1	0

Table 12: Device utilization summary for multiplierless kernel of multiclass SVM

Device: VIRTEX7 xc7vx485t	Utilized	Available	% Utilization
No. of Slice LUTs:	33360	303600	11%
No. used as Logic:	33360	303600	11%
No. with an unused Flip Flop:	33360	33360	100%
No. with an unused LUT	0	33360	0%
No. of bonded IOBs:	28	600	4%
No. of DSP48E1	0	2800	0%

Table 13: On- Chip Power Summary for multiclass SVM

On- chip	vector product(W)	Multiplierless(W)
Clock	0.010	0.002
Logic:	0.126	0.236
PLL	0.114	0.114
Others	0.332	0.332
BRAM	0.000	0.000
I/Os	0.766	0.766
DSPs	0.163	0
Device Statistics	0.253	0.253
<b>Total</b>	<b>1.764</b>	<b>1.703</b>

### 4.3 Multiclass classifier hardware implementation

This design is implemented in XILINX 7vx485tffg1157-2. The primitive and black box uses of LUTs, DSP4s and clocks for multiclass classifier using multiplierless kernel is given in Table 11. The design utilization summary using multiplierless kernel is given in Table 12. The synthesis report estimates the resources used for every design implemented in Xilinx. The power is measured using XILINX POWER ESTIMATOR (XPE) 14.1. This version is used specifically for the 7 series XPE. The on chip power summary report of both the vector product kernel module and multiplierless kernel for the multiclass classification is given in Table 13.

From Table 12, we can analyze that the percent-

Table 14: Power reduction using multiplierless kernel compared to vector product kernel

Binary linear	Binary nonlinear	Multiclass
1%	2.7%	3.5%

age utilization of LUT and slice logic is very high in case of multiplierless kernel. This is because we have 74 PEs and in case of multiplierless kernel each PE carries all logics to perform multiplierless vector operation. On chip power summary of multiclass SVM is demonstrated in Table 13. Here as the resource utilization (LUT) is higher in multiplierless kernel so the logic consumes higher power in multiplierless kernel than the vector product kernel. All the other power consumption factors consumes almost same power except dsp. As in case of vector product kernel the dsp power consumption is much higher due to use of large number of dsp units, the total power consumption decreases from 1.764 to 1.703 using multiplierless kernel. So we have reached to our goal that is around 3.5% reduction in power is achieved in case of multiclass SVM. % reduction in power using proposed kernel for all the classification results are listed in Table 14. From this table we can see that the power decrease is high in case of binary nonlinear and multiclass kernel which are mainly used for real-time applications.

## 5 Conclusion

SVM is efficient in classification problems and has numerous application in object detection, computer vision and image processing. It has been effectively implemented in software. But the hardware implementation of this continues to be challenging. Here, we have proposed the design of multiplierless kernel function which is suitable for binary and multiclass problems with both low and high dimension data. The proposed multiplierless kernel is used in case of binary linear, binary nonlinear and multiclass classification. All of these classification problems showing successful results in classifying the data. We have also implemented the above three different classification problems in hardware using multiplierless kernel module. Comparative analysis of all these three classification problems are done using multiplierless kernel and using conventional vector product kernel regarding resource utilization. We are also able to reduce the power requirement of the hardware design of all the three classification problems discussed here using the proposed multiplierless kernel compared to the vector product kernel.

**Acknowledgements:** This work is supported by Dr. Kandarpa Kumar Sarma, Manash Pratim Sarma, Dept of ECE, Gauhati University. I am highly indebted to all of them for their constant supervision and support in completing the project.

*References:*

- [1] J. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition”, burges@lucent.com, 1998.
- [2] S. R. Gunn, “Support Vector Machines for Classification and Regression”, *University of Southampton*, 1998.
- [3] R. Reyna, D. Esteve, D. Houzet and M.-F. Albenge, “Implementation of the SVM neural network generalization function for image processing” *Proceedings Fifth IEEE International Workshop on Computer Architectures for Machine Perception*, pp. 147 -151, 2000.
- [4] D. Anguita, A. Boni and S. Ridella, “A digital architecture for support vector machines: theory, algorithm, and FPGA implementation”, *IEEE Transactions on Neural Networks*, vol. 14, no. 5, pp. 993 - 1009, Sept. 2003.
- [5] I. Biasi, A. Boni and A. Zorat, “A reconfigurable parallel architecture for SVM classification”, *IEEE International Joint Conference on Neural Networks*, vol. 5, pp. 2867 - 2872, July 2005.
- [6] S. Cadambi, I. Durdanovic, V. Jakkula, M. Sankaradass, E. Cosatto, S. Chakradhar and H. Graf, “A massively parallel FPGA-based coprocessor for support vector machines”, *17th IEEE Symposium on Field Programmable Custom Computing Machines*, pp. 115 -122, Apr. 2009.
- [7] C. Kyrkou and T. Theocharides, “SCoPE: Towards a systolic array for SVM object detection”, *Embedded Systems Letters, IEEE*, vol. 1, no. 2, pp. 46 -49, Aug. 2009.
- [8] S. Kim, S. Lee and K. Cho, “Design of High-Performance Unified Circuit for Linear and Non-Linear SVM Classifications”, *Journal of semiconductor technology and science*, vol.12, no. 2, pp-162-167, June 2012.
- [9] R.A. Patil, G. Gupta, V. Sahula, A. S. Mandal, “Power aware Hardware prototyping of multi-class SVM classifier through Reconfiguration”, *25th International Conference on VLSI Design*, pp-62-67, 2012.
- [10] Y. Goldberg and M. Elhadad, “splitSVM: Fast, space-efficient, non-heuristic, polynomial kernel computation for NLP applications”, *In Proceedings of the 46st Annual Meeting of the Association of Computational Linguistics*, 2008.
- [11] J. O. Coleman, A. Yurdakul, “Fractions in the Canonical-Signed-Digit Number System”, *Conference on Information Sciences and Systems, The Johns Hopkins University*, March 2001.
- [12] K. K. Parhi, *VLSI digital signal processing system*. 1st ed. New Delhi: Wiley India (P.) Ltd., 2012.
- [13] D. Knuth, *The Art of Computer Programming*, Vol 2, 3rd ed, Addison Wesley, 1997.
- [14] B. Mandal, M. P. Sarma, K. K. Sarma, “Design of Systolic array based Multiplierless SVM Classifier”, *IEEE International Conference on Signal Processing and Integrated Networks*, pp-35-39, Noida, Feb 2014.