International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/IJSRCSEIT

FPGA Implementation of Classification Based on SVM For Smart Grid Application

[1] Ms. T. Tamileselvi, [2] K.Kanagaraj, [3] M. Korkaimaran, [4] M. Surya Rao
[1] Associate Professor, [2,3,4] UG Scholar
[1,2,3,4]Department of ECE, Jerusalem College of Engineering, Pallikaranai, Tamil Nadu, Chennai, India

Article Info

Publication Issue :
Volume 9, Issue 1
January-February-2023

Page Number : 72-79

Article History
Accepted: 01 Jan 2023
Published: 15 Jan 2023

ABSTRACT

In order to increase predictive accuracy, Support Vector Machines (SVMs), a common classification and regression prediction tool, employ supervised machine learning theory. The field programmable gate array (FPGA) implementation of a classification system using a support vector machine is the main emphasis of this study. The FPGA-based two-class SVM classifier can rapidly classify data due to the powerful parallel computation capability it offers. Depending on the classification's dimensions, the system operates in either a linear or non-linear fashion. The system would help in the classification system which is efficient at quickly classify data and is a promising method for enhancing communication security in the Smart Grid.

Keywords - FPGA devices , support vector machine and smart grid.

I. INTRODUCTION

Over the past 20 years, machine learning has led to the development of modern classification approaches such as support vector machines (SVMs), boosted decision trees, regularised logistic regression, neural networks, and random forests[1]. One of the numerous techniques that have proved effective in addressing information analysis issues, particularly those involving data classification and regression, is support vector machines. SVMs have been successfully applied in the real world to address complicated information analysis problems, as opposed to many other methods[2]. The SVMs method processes the data by minimising an empirical risk in a well-posed and consistent manner, which is based on regularisation theory.

SVM classification, a kind of large-margin classifier, uses vector space as its foundation and seeks to construct a judgement border between two classes that is as remote from any point in the training data as is practical. Since it is easier to separate the input data in this higher-dimensional feature space, the low-dimensional training data is projected onto it as the foundation of SVMs[3]. Additionally, even when training data cannot be separated linearly in the low-dimensional feature space, this projection might allow for linear separation in the high-dimensional space using kernel functions.

The field of artificial intelligence known as "machine learning" is concerned with developing techniques and algorithms that enable computers to learn. To aid the machine's ability to learn and perform tasks seen in the real world, many algorithms have been developed. Its objective is to apply machine learning theory to solve real-world issues. When used for pattern identification, SVM becomes well-known; in a handwriting recognition challenge, it offers accuracy equivalent to sophisticated neural networks with detailed features. It is also widely used by many to reduce an empirical risk in a well-posed and consistent manner on the basis of regularization theory. A type of large-margin classifier, SVM classification is based on vector space and aims to establish a decision border between two classes that are as far away from any point in the training data as possible.

II. MACHINE LEARNING

Utilizing sample data or prior knowledge, machine learning involves programming computers to optimize a performance criterion. We have a model, as shown in Fig. 1, that is defined up to a certain point, and learning is the application of a computer programme to optimize the model's parameters using training data or prior knowledge[4]. The model may be descriptive to learn from the data or predictive to make future predictions. While working at IBM in 1959, Arthur Samuel, a pioneering American in the fields of artificial intelligence and video games, invented the phrase "Machine Learning." "The field of research that gives computers the ability to learn without being explicitly programmed," he said of machine learning. However, there is no universally accepted definition of machine learning. Different authors define the term differently. The machine learning types are classified as supervised and unsupervised machine learning which is discussed in next session [5].

III. TYPES OF MACHINE LEARNING

(i) Supervised learning

The algorithm generalises to reply appropriately to all potential inputs based on a training set of examples with the correct responses (targets). This is also known as imitation learning[6]. A function that maps an input to an output is learned through supervised learning using sample input-output pairs.

Each example in the training set for supervised learning is a pair that consists of an input item (usually a vector) and an output value. An algorithm for supervised learning examines the training data and generates a function that can be applied to mapping fresh examples. The function will appropriately identify the class labels for instances that aren't visible in the best case scenario. Regression and classification issues are both supervised learning issues. There are numerous supervised learning algorithms available, each having advantages and disadvantages.

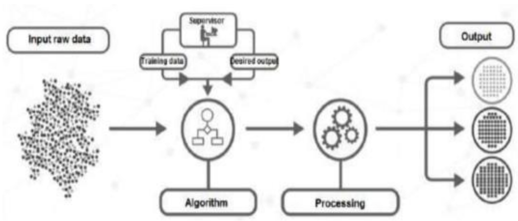

Fig 1. Machine learning process

(ii) Unsupervised learning

Instead of providing correct answers, the algorithm instead looks for patterns in the inputs so that inputs with similar characteristics can be grouped together. Density estimation is an approach to unsupervised learning that uses statistics. Drawing conclusions from datasets of input data without labelled replies is possible using unsupervised learning, a sort of machine learning method[7]. The observations in unsupervised learning algorithms do not involve classification or categorization. Since there are no output values, no function estimation is possible.

Since the samples provided to the learner are unlabeled, it is impossible to assess the accuracy of the structure that the algorithm produces. Cluster analysis, which is employed for exploratory data analysis, is the most popular unsupervised learning technique. The most common unsupervised learning method is cluster analysis, which is used for exploratory data analysis to find hidden patterns 14 or grouping in data

(iii)

einforcement learning

Between supervised and uncontrolled learning, this falls. Although the algorithm is informed when the solution is incorrect, it is not informed how to make it right. It must investigate and test out several options before figuring out how to provide the correct response. Because the monitor only assigns a score and makes no suggestions for improvement, reinforcement learning is sometimes referred to as learning with a critic[8]. The issue of persuading an agent to behave in the real world in a way that maximises its rewards is known as reinforcement learning. Unlike most forms of machine learning, a learner (the software) is not given instructions on what to do; rather, it must experiment with several options to determine which ones produce the greatest rewards[9]. In the most interesting and challenging cases, actions may affect not only the immediate reward but also the next situations and, through that, all subsequent rewards.

IV. SUPPORT VECTOR MACHINE CONCEPTS AND APPLICATION

Support Vector Machines, Bayes Point Machines, Kernel Principal Component Analysis, and Gaussian Processes are just a few examples of kernel-based techniques that have significantly advanced machine learning algorithms. A class of supervised learning techniques known as support vector machines (SVMs) 12 can be used for classification or regression[10].

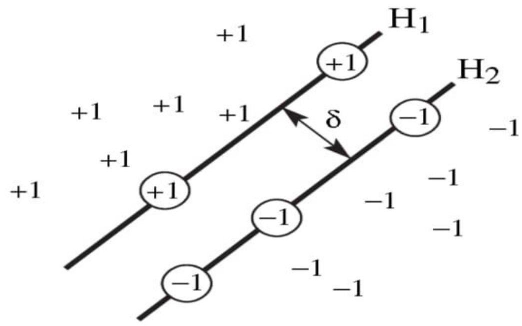

Fig2. Separation using hyperplane

R

An SVM identifies the single hyperplane with the greatest margin for any given set of two-class objects (denoted with in Fig 2) While the border of the hyperplane H2 is defined by items of class -1, that of the hyperplane H1 is defined by objects of class +1. The hyperplanes H1 and H2 are defined by two objects from class +1 and three objects from class -1, respectively. Support vectors are these objects, which are shown in Fig. 2 as circles[11].

The solution to a classification problem is represented by the support vectors that establish the maximum margin hyperplane, which is a unique property of SVM. SVM models were initially created to categorise classes of items that could be separated linearly. The example is shown in Fig. 3. Finding a line that completely divides these two-dimensional objects into their two classes (class +1 and class -1), which they belong to, is simple. SVM can be used to differentiate classes that a linear classifier is unable to (Fig.3, left). In these circumstances, nonlinear functions known as feature functions are used to map the object coordinates into a feature space. The feature space is a high-dimensional space in which the two classes can be separated with a linear classifier (Fig.3, right).

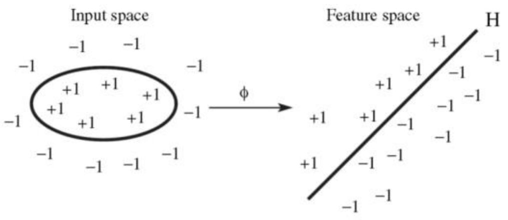

Fig3. Linear separation in feature space

.

The nonlinear feature function, as shown in Figs. 3 and 4, merges the input space—the original coordinates of the objects—into the feature space, which may even be infinite in size. It is not practicable to directly apply feature functions while computing the classification hyperplane due to the high dimension of the feature space. Instead, specialised nonlinear functions known as kernels are used to compute the nonlinear mapping caused by the feature functions. Since the answer to the classification problem is a weighted average of kernel functions assessed at the support vectors, kernels have the advantage of working in the input space.

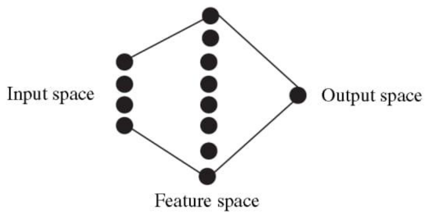

Fig4. : Support vector machines map the input space into a high-dimensional feature space

Consider the patterns in Table 1.1 to demonstrate how SVM can be used to train nonlinear classifiers. This artificial dataset of two-dimensional patterns was created to study the SVM classification algorithm's characteristics. Class +1 patterns are denoted in all figures by +, while class -1 patterns are denoted by black dots. The SVM hyperplane is shown as a continuous line, while its edges are depicted by dotted lines. Support vectors belonging to the class +1 are denoted by a plus sign (+) inside a circle, whereas those belonging to the class -1 are denoted by a black dot inside a circle. Figure 5a displays the partitioning of the dataset from Table 1.1 using a linear kernel.. It is obvious that a linear function is not adequate for this dataset because the classifier is not able to discriminate the two types of patterns; all patterns are support vectors. A perfect separation of the two classes can be achieved with a degree 2 polynomial kernel (Fig.5b).

Table 1.1. Linearly Non-separable Patterns used for the SVM Classification Models in Fig 3.

Pattern	x1	x2	Class
1	2	4.5	1
2	2.5	2.9	1
3	3	1.5	1
4	3.6	0.5	1
5	4.2	2	1
6	3.9	4	1
7	5	1	1
8	0.6	1	-1
9	1	4.2	-1
10	1.5	2.5	-1
11	1.75	0.6	-1
12	3	5.6	-1
13	4.5	5	-1
14	5	4	-1
15	5.5	2	-1

Six support vectors total, three from class +1 and three from class -1, make up this SVM model. It is possible to forecast the class membership for new patterns using these six patterns, which also serve to define the SVM model. The SVM model may be defined without the four patterns from class +1 located in the space region bounded by the +1 margin and the five patterns from class -1 located in the space region bounded by the -1 margin. As such, they can be removed from the training set without affecting the SVM solution.

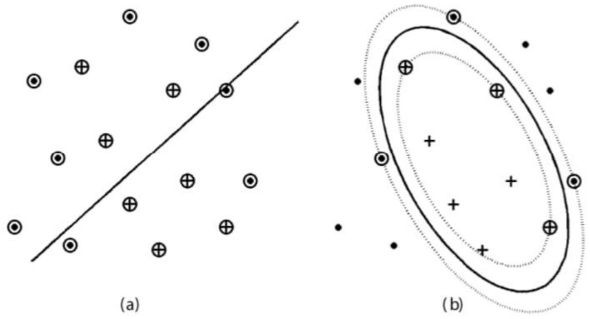

Fig5. SVM classification models for the dataset from Table 1.1

A. LINEAR SUPPORT VECTOR MACHINE

Multi-class classifications can be divided into two-class classification units, and by substituting Kernel Functions for inner product calculations, non-linear classification problems can be resolved.

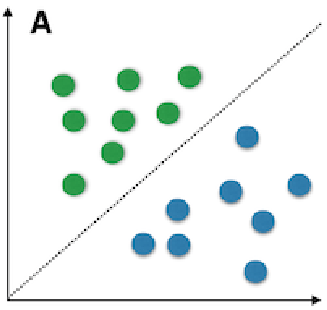

Fig5. Linear classifier

The Linear SVM classifier is used for linearly separable data, as shown in Fig. 5. This indicates that if a dataset can be divided into two classes using just one straight line, it is considered to be linearly separable data.

B. NON-LINEAR SUPPORT VECTOR MACHINE

If a dataset cannot be classified using a straight line, then the data is referred to as linear data and the classifier. Non-Linear SVM is depicted in Fig 6.

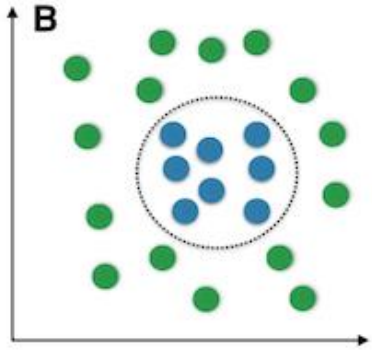

Fig.6 Non Linear Classifier

V. MODERN FIELD PROGRAMMABLE GATE ARRAY DEVICE

A semiconductor device that can be programmed after manufacturing is the Field Programmable Gate Array (FPGA). An FPGA is referred to as "field-programmable" because it allows you to programme product features and functions, adapt to new standards, and reconfigure hardware for particular applications even after the product has been installed in the field, as opposed to being limited to any predetermined hardware function. Any logical operation that an application-specific integrated circuit (ASIC) may carry out can be implemented using FPGA, although many applications benefit from the flexibility of updating the functionality after shipping.

Instead of using equations and functions, we must create a compatible formulation of the algorithm to solve the SVM classification problem with PFGA.

Linear SVM classification algorithm.

```
set matrix x=[ Test sample ];
    matrix X=[ Training samples ];
    matrix Y=[ Class identity ];
    matrix B=Y';
for i=1 to n,
for j=1 to n,
multiply matrix X(i,:) by matrix X(j,:)'
set matrix A(i,j)=X(i,:)*X(j,:)';
end
end
for k=1 to n,
    for l=1 to n,
    do multiply matrix A(k,l) by matrix Y(l)
    set matrix A(k,l)=A(k,l)*Y(l);
    end
    end
divide matrix A by matrix B
set matrix C=A\B;
for m=1 to n,
     multiply matrix C(m) by matrix Y(m) and matrix X(m,:)
     set matrix Z(m,:)=C(m)*Y(m)*X(m,:);
     end
set matrix W=[sum of the first column of matrix Z, sum of the second column of matrix
Z];
calculate parameter b=Y(1)-W*X(1,:)';
build the classification function G=W*x'+b
classification result G = Ans
```

$$\begin{cases} \operatorname{sgn}\{G\}=1, & \mathbf{X}\in C_1 \\ \operatorname{sgn}\{G\}=-1, & \mathbf{X}\in C_2 \end{cases}$$

The 18 bit signed fixed-point adders and multipliers with 1 sign bit, 7 bits before the decimal point, and 10 bits after are the computational modules created for the system. The computational units are constructed using a fixed point package. The VHDL 1076.3 numeric std package serves as the foundation for the fixed-point math packages, which utilise its signed and unsigned arithmetic.

As a result of the numeric std package's strong support from simulation and synthesis tools, they are quite effective. Unsigned fixed point type "ufixed" and signed fixed point type "sfixed" are two new types that are defined by the package.

Because of the fixed-bit restriction, the computational accuracy is slightly reduced when fixed-point packages are applied to specified computing units.Any data with accuracy higher than $(½)^{10}$ can't be processed and expressed precisely.

Non-linear SVM Algorithm:

```
set matrix x=[ Test sample ];
    matrix X=[ Training samples ];
    matrix Y=[ Class identity ];
    matrix B=Y';
for i=1 to n,
for j=1 to n,
calculate classification function parameters using matrix X and Gaussian radial basis
function and give the result to matrix
A(i,j)=exp(-((X(i,1)-X(j,1))^2+(X(i,2)-X(j,2))^2)/(2*(0.6)^2));
end
end
for k=1 to10,
    for l=1 to 10,
    multiply matrix A(k,l) by matrix Y(l) and give the result to matrix A(k,l);
    end
    end
dived matrix A by matrix B and give the result to matrix C;
for q=1 to 10,
calculate C(q)*Y(q)*exp(-((X(q,1)-X(1,1))^2+(X(q,2)-X(1,2))^2)/(2*(0.6)^2)) and
give the result to h(q);
end
calculate the sum of all the elements in matrix h and give the result to matrix H;
calculate parameter b=Y(1)-H;
for p=1 to 10,
calculate matrix
g(p)=C(p)*Y(p)*exp(-((X(p,1)-x(1))^2+(X(p,2)-x(2))^2)/(2*(0.6)^2));
end
calculate the sum of matrix g and give the result to matrix W;
build the classification function G=W+b
```

$$\begin{cases} \operatorname{sgn}\{G\}=1, & \mathbf{X}\in C_1 \\ \operatorname{sgn}\{G\}=-1, & \mathbf{X}\in C_2 \end{cases}$$

We create a non-linear SVM classification technique by substituting the Kernel Function for the dot product. For an exponential function, neither fixed point nor floating point calculations can be implemented on an FPGA effectively. Massive exp computing does not completely exploit the benefits of parallel computing. We created a table-driven exponential function calculation module for the suggested approach to carry out the actual exp calculation. The table-driven exponential module's use resulted in significant PFGA computer resource savings, and the simulation test's results indicate that the accuracy is trustworthy. By using processing units akin to those used in a linear approach, the matrix calculation is also carried out in parallel. In our design, the Gaussian radial basis function's parameter is set to 0.6.

VI. RESULTS

We created four distinct two-class linear datasets (datasets A, B, C, and D) for the linear SVM architecture in order to evaluate the classifier. Each dataset has a 400-point size, of which 20 will be used for SVM training and 50 for testing.

The test results provided by the classification system are satisfactory, and the time spent on each dataset's cumulative results satisfies our design requirements. However, depending on the volume of training data and the density of the support vectors, the accuracy of the results may fluctuate slightly.

Table2: Test results of Linear model

	Accuracy	Recognition rate
Model A	99.58%	100%
Model B	99.63%	98%
Model C	99.62%	100%
Model D	99.61%	100%

Four different 400-size nonlinear datasets are constructed for the non-linear SVM classification architecture in order to evaluate the non-linear classifier. Each dataset has a training size of 20 and a testing size of 40. The overall outcomes are also acceptable. The computation error is less than the linear design, at about 0.041%.

Table 3 Test results of Non Linear Model

	Accuracy	Recognition rate
Model A1	99.95%	97.5%
Model B1	99.96%	100%
Model C1	99.94%	100%
Model D1	99.99%	95%

VII. CONCLUSION

The SVM classification system presented in this paper is FPGA-based and can classify data quickly. The results of the implementation demonstrate that the quick, two-class, high-accuracy, planned FPGA implementation of the SVM classification system performs satisfactorily. The specified requirements are satisfied by the SVM classifier's performance as a quick recognition classification system. It is highly encouraging for future work that smart metres integrated with SVM classifiers can offer quick intrusion detection to safeguard the entire secure communication system, much like a firewall.

VIII. REFERENCES

[1]. Markos Papadonikolakis, Christos-Savvas Bouganis, “Novel Cascade FPGA Accelerator for Support Vector Machines Classification,” IEEE Transactions on Neural

[2]. Networks and Learning Systems, Vol. 23, No. 7, July 2012

[3]. M. Papadonikolakis and C.-S. Bouganis, “A scalable FPGA architecture for non-linear SVM training,” in Proc. Int. Conf. FPT Technol., Dec. 2008, pp. 337–340.

[4]. 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,” in Proc. 17th IEEE Symp. Field Programm. Custom Comput. Mach., Apr. 2009,

[5]. D. Anguita, A. Boni, and S. Ridella, “A digital architecture for support vector machines: Theory, algorithm, and FPGA implementation,” IEEE Trans. Neural Network., vol. 14, no. 5, pp. 993–1009, Sep. 2003

[6]. F. Khan, M. Arnold, and W. Pottenger, “Hardware-based support vector machine classification in logarithmic number systems,” in Proc. IEEE Int. Symp. Circuits System., vol. 5. May 2005, pp. 5154–5157.

[7]. K. Irick, M. DeBole, V. Narayanan, and A. Gayasen, “A hardware efficient support vector machine architecture for FPGA,” in Proc. Annu. IEEE Symp. Field-Programm. Custom Comput. Mach., Apr. 2008

[8]. B. Catanzaro, N. Sundaram, and K. Keutzer, “Fast support vector machine training and

classification on graphics processors," in Proc. 25th Int. Conf. Mach. Learn., 2008, pp. 104–111.

[9]. A. Carpenter. (2009). CUSVM: A Cuda Implementation of Support Vector Classification and Regression [Online]. Available: http://patternsonascreen.net/cuSVM.html

[10]. NVidia. (2008). NVIDIA CUDA Compute Unified Device Architecture, Programming Guide, Santa Clara, CA [Online]. Available: http://www.nvidia.co.uk/cuda.

[11]. M. Papadonikolakis, C.-S. Bouganis, and G. Constantinides, "Performance comparison of GPU and FPGA architectures for the SVM training problem," in Proc. Int. Conf. Field-Programm. Technol., 2009

[12]. T.Tamilselvi, Dr.V.Rajendran, "Comparative Study of SVM and KNN Machine Learning Algorithm for Spectrum Sensing in Cognitive Radio", 2022, in Intelligent Communication Technologies and Virtual Mobile Networks, Lecture Notes on Data Engineering and Communications Technologies 131, https://doi.org/10.1007/978-981-19-1844-5_41

Cite this article as :

Ms. T. Tamileselvi, K. Kanagaraj, M. Korkaimaran, M. Surya Rao, "FPGA Implementation of Classification Based on SVM For Smart Grid Application", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 9 Issue 1, pp. 72-79, January-February 2023.
Journal URL : https://ijsrcseit.com/CSEIT2390118