IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

Classification of Pseudo Random Numbers Using Support Vector Machine Classifier

Preeti Meena[1], Malti Bansal[2]

Department of ECE, Delhi Technological University, Delhi, India[1, 2]

Abstract: The paper studies about the randomness and its application in the field of cryptography. It is already known to have randomness having applications in the fields of gaming, gambling, statistics etc. But, in applications such as cryptography, randomness has a vital role. As with the use of these random numbers only, a highly secure message can be formed. So, Pseudo Random numbers classification is proposed in this paper so that sources of these random numbers can be classified for the enhancement in the security level. The result shows good accuracy of classification of these random numbers i.e. classified using multi-SVM.

Keywords: Cryptanalysis, Multi-SVM, PRNGs, TRNGs.

I. INTRODUCTION

In this world, randomness is the dearth of certainty in an event. It can be easily understood with the help of randomness that proper random numbers generation is a quite complicated process. Random numbers are helpful in various purposes which include creating files keys for encryption, creating a representation for selecting the random trials from large datasets. A random sequence for any event has no particular order and does not follow a logical pattern.

Randomness has found its applications in the fields of art, science, statistics, cryptanalysis, gaming etc. In the field of cryptanalysis, goal is to make a message as hard to crack as possible. It is done by disguising the parameters used to encode the message from the context in which it is conceded. For example, randomness, in randomized trials, are helpful in testing the hypotheses. Pseudo Random numbers are useful in video games such as video poker. In 1946, [1] introduced a method to generate Pseudo Random numbers. In this method, middle digit of the previous number takes the place of the next successive number. First of all they took a kernel value, and calculated the square in which they nominated the middle digits of that number and took it as the kernel value for next successive Pseudo Random number. This way, middle digits of previous numbers act as the kernel value for the successive number. In 1949, [2] proposed a method named Linear Congruential Generator (LCG). This algorithm comes out as the best algorithm for the generation of Pseudo Random numbers. It is very fast and easy to comprehend at the same time. Its execution is also very informal. In this, with the help of a simple linear equation, they attained a Pseudo Random sequence. The source of the equation was the modulo arithmetic. Modulo arithmetic is a very useful tool in which numbers wrap around after reaching a certain value. Quadratic congruential generator 1 and Quadratic congruential generator 2, uses a quadratic equation, are very greatly based on the idea of Linear congruential generator as these two do not use the linear equation. [3] also generates another Pseudo Random number generator, named Blum Blum Shub, that was proposed in 1986. These Pseudo Random numbers generated using various different methods can increase their number of applications if the source of that random number is identified. Support Vector Machine is a model which is derived from a theory known as statistical learning [4]. Compared to Neural Network (NN) approaches [5], SVMs exhibit higher generalization capability, robustness, lower efforts are required for classifier selection during the training period [6] and finest solution is obtained by this algorithm. [7] proposed a technique which performs classification of multi-classes problem (known as multiclass SVM) into the single one, rather into binary classification of multiple sets. Now, through this paper, an experiment has been done for separating the different kinds of Pseudo Random numbers generated from different sources. For this, three set of Pseudo Random numbers are selected with different sources, each representing a separate class. Then using the model named SVM, classification of these numbers are done and results are represented in the form of an image output showing three classes of Pseudo Random numbers which are well separated using the SVM.

The paper is organised as follows. Section II provides details about definitions and explanations of the methods used. Section III provides the process, explained using algorithm, of the proposed method. Section IV contains the classification results followed conclusion in Section V.

II. PROPOSED METHOD

A. True Random Numbers Generator (TRNG)

In the applications like cryptography, sources for generating the random numbers should be of high quality so that a high quality design may produce. There are so many good quality sources exists for the generation of true random numbers, at present, to design a cryptographic system. But, unfortunately, cryptographic analysis is done with only few of the existing true random number

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

generators. This is because a design made using TRNGs produces the sequence of numbers of such kind that these numbers show a definite level of association due to its bandwidth and temperature drift limitations.
So, due to issues in manipulating the TRNGs, there is difficulty in designing such a system. TRNGs also create deterministic disturbances in the cryptographic system design. Solution to this problem is to design such a system which can use analogue randomness sources. Therefore, due to design flaws occurring using TRNGs and lack of random data in TRNGs, Pseudo random numbers generators came in use.

B. Pseudo Random Numbers Generators (PRNG)
Theories of researches on Pseudo Random numbers with the results being so positive that Pseudo Random numbers generated using modern algorithms look exactly as they are really random. According to the meaning of the word 'Pseudo', Pseudo Random numbers are not as such as we might presume. Essentially, PRNGs are the algorithms which utilize simple pre-calculated data and mathematical formulaes [11] to generate the random numbers. Linear congruential is a good design of PRNGs.
The linear Congruential Generator (LCG) is defined by using a recurrence relation:

$$Y_{k+1} = (\alpha.Y_k + \gamma).(|r|) \qquad (3)$$

where,
α, 0 < α < r = multiplier
γ, 0 <= γ < r = incremental
r, r > 0 = modulus
Y_0, 0 <= Y_o < r = seed value
And, Y is the sequence of Pseudo Random values

$$Gcd(\gamma, r) = 1 \qquad (4)$$

β = (α - 1) is multiple of q for every q dividing r
β is multiple of 4 if r is multiple of 4

Another PRNG that exists is Quadratic Congruential Generator i.e., shown in the equation

$$Y_{k+1} = (\alpha Y_k^2)(|r|) \qquad (5)$$

Quadratic congruential generator 2 is defined as,

$$Y_{k+1} = (\alpha Y_k^2 + \beta Y_k + \gamma)(|r|) \qquad (6)$$

Blum Blum Shub is another PRNG which follows the equation for PRNG as:

$$Y_{k+1} = Y_k^2(|r|) \qquad (7)$$

Here, Y_r = seed value
Y_{k+1} = Pseudo Random number generated
r = uv, where u and v be the two prime numbers of large values. In this algorithm, output is derived from Y_K at each step. Then the output is the least significant bit of Y_{K+1}.

Blum Blum Shub follows the following steps:
1. Generate u and v as two big prime numbers.
2. k := u · v
3. Choose t belongs to R [1, k − 1], the random seed. Choose a random number s, such that neither u nor v is a factor of t.
4. $Y_0 := t_2 \pmod k$
5. The sequence is defined as $Yj = Y_{2j-1} \pmod k$ and $Z_j :=$ parity(Y_j).
6. The output is $Z_1, Z_2, Z_3, \ldots$.

PRNGs are efficient [12] which means that they can generate the random numbers in a short extent. These are deterministic also as certain sequence can be reproduced at the future stage if the primary point of the structure is known. If determinism is handy, same sequence is to be repeated at further stage. Efficiency can also be considered as a distinguishing characteristic if there are large number of pseudo random numbers are present. PRNGs also have one more property which is hardly desirable and i.e. periodic nature [13]. This property states that the sequence obtained should be repeatable. At present, modern algorithms of PRNGs are periodic with such a period that it can be ignored for most practical cases as it is so long.

As it is beneficial to have a sequence which can be easily repeatable, hence, the above stated characteristics of PRNGs make them a very much suitable for the applications where randomness is the basic necessity. Among such applications, popular ones includes simulation and modeling applications but are not suitable for the applications such as gambling and data encryption, where the numbers are really unpredictable.

C. Comparison of PRNGs with TRNGs
In comparison with PRNGs [14], the characteristics of TRNGs are quite different. These characteristics are helpful in distinguishing both. Firstly, TRNGs are ineffective as compared to PRNGs because former takes considerably a very large time to generate the numbers. They are also known by the name of non-deterministic random number generators because it is not possible with the help of TRNGs that random numbers sequence can be reproduced.

Athough, by chance, same numbers sequence may, of course, occur several times. TRNGs is non-periodic. The basic differences [15] between PRNGs and TRNGs are very easy to understand because PRNGs produce random numbers by using pre-calculated data and mathematical formulaes. An outlook is shown in the Table I in which the characteristics of the two types of random number generators are compared.

Even the numbers generated from TRNG seems random, but they are really predetermined. TRNGs are considered more suitable, approximately, in such applications in which PRNGs are failed to find their applications. These includes security in data transfer, gambling, games etc. However, less efficient and non-deterministic nature of TRNGs doesn't allow such generated random numbers to produce satisfactory results for the applications like simulation and modeling. This is because TRNG is not able to generate the amount of data which is required for these applications.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

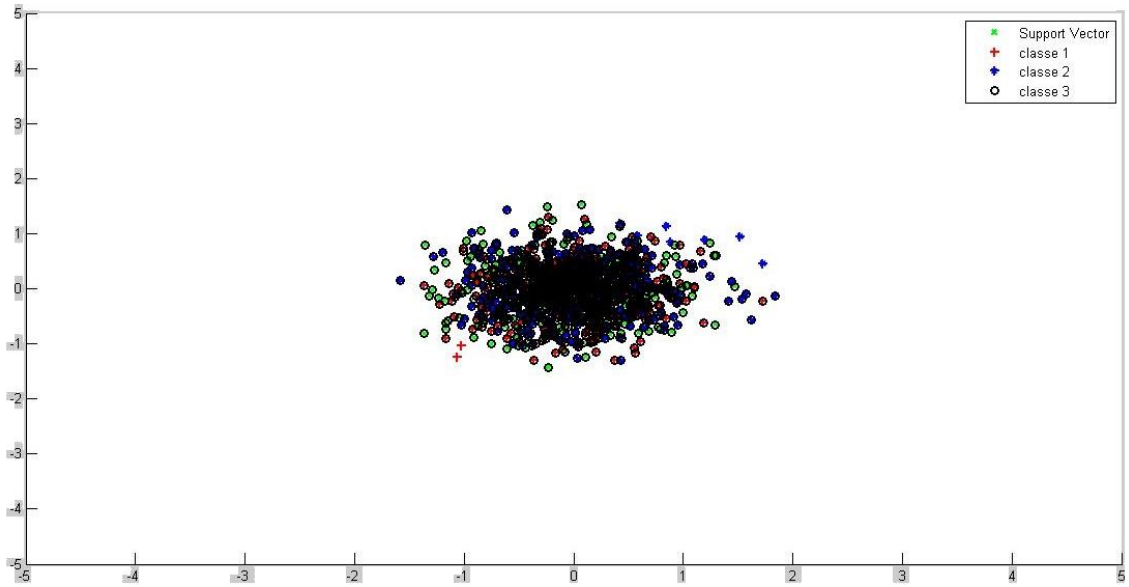

Fig. 1. Mixture of three classes Pseudo Random Numbers

TABLE I COMPARISON OF PRNGS AND TRNGS

Characteristics	True RNGs	Pseudo RNGs
Efficient	Less	more
Nature	Random	Deterministic
Repeatablity	Non-repeatable	Repeatable

Table II shows the various fields of applications and generators suitable in each field.

TABLE II SUITABLITY OF GENERATOR : APPLICATION-WISE

Application	Most Suitable Generator
Generation of keys for data encryption	True RNG
Random Sampling (e.g., medicine selection)	True RNG
Sports and Gambling	True RNG
Lotteries and Draws	True RNG
The Arts	Varies
Simulation and Modelling	Pseudo RNG

Hence, according to its application in the simulation and modelling, Pseudo Random numbers are used during this research. Three different types of Pseudo Random numbers generated are mixed and classified again separately using a classifier. The classifier is explained in the next sub-section.

D. Classifier

In the classification of the pseudo random numbers generated using different sources, SVM is a classifier which is used. It is a supervised learning model [16] as it is firstly trained using the associated learning algorithms with the labelled data. Then with the help of this trained model, tested data is analyzed. These are exploited in the regression analysis. SVMs can be helpful in various applications. These includes text categorization, hand-written characters recognition etc. Experimental results shows, SVM achieves higher accuracy than other traditional models. SVMs also find its application in the field of medical sciences with nearly 80% of accuracy in classifying the samples accurately. For this, a set of training random numbers, each marked for one of the classes is used to train the SVM. Then the pseudo random numbers are tested on this trained SVM and the random numbers are classified into the classes to which they belong. SVM can easily with efficient results performs non-linear classification using kernel trick in which dimensionality is increased for the feature space. Let y_i, j =1, 2, . . . ,K, be the pseudo random numbers for the training Y. Numbers taken belong to one of the three sources (or classes). These random numbers are supposed to be separable linearly. Now, the objective is to design a hyperplane such as

$$h(y) = \omega^T y + \omega_0 = 0 \tag{8}$$

Then the separation of a support vector from the hyperplane is given by

$$D = \frac{|h(y)|}{\|\omega\|} \tag{9}$$

Now, ω, ω_0 are selected such that the value of h(y) should be at the support vector points. This is equivalent with

1. Having a margin of

$$\frac{1}{\|\omega\|} + \frac{1}{\|\omega\|} = \frac{2}{\|\omega\|}$$

DOI 10.17148/IJARCCE.2016.55177

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

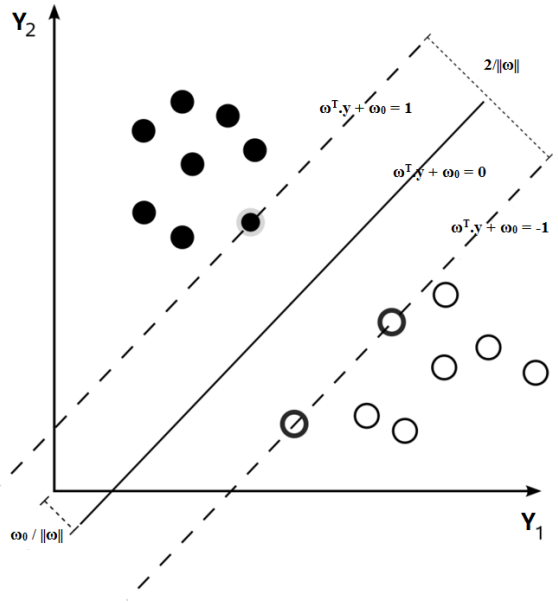

Fig. 2. Linear Classifiers

2. Requiring that

$$\omega^T y + \omega_0 \geq 1, \forall y \in \omega_1 \quad (10)$$

$$\omega^T y + \omega_0 \geq -1, \forall y \in \omega_2 \quad (11)$$

Now, for each value of y_j, we represent the corresponding class by z_j. Hence, for this, calculate ω, ω_0 so that:

$$\text{minimize } F(\omega, \omega_0) \equiv \frac{1}{2}\|\omega\|^2 \quad (12)$$

subject to

$$z_j(\omega^T y + \omega_0) \geq 1 \quad , j = 1, 2, \ldots, \quad (13)$$

This is done as minimizing (12) will make the separation between support vector and hyperplane maximum. Some constraints need to be followed by the SVM. Here are the Karush–Kuhn–Tucker (KKT) conditions [17] that the minimizer has to satisfy are as follows:

$$\frac{\partial}{\partial \omega} L(\omega, \omega_0, \eta) = 0 \quad (14)$$

$$\frac{\partial}{\partial \omega_0} L(\omega, \omega_0, \eta) = 0 \quad (15)$$

$$\eta_j \geq 0 \quad , j = 1, 2, \ldots, K \quad (16)$$

$$\eta_j[z_j(\omega^T y + \omega_0) - 1] = 0 \quad , j = 1, 2, \ldots, K \quad (17)$$

More formally, SVM will design a hyperplane in a higher dimensional space using the kernel trick which can be exploited for the classification purpose. But here, three classes of random number are present generated from different sources and no need of kernel function is there during this classification. Here, multiclass SVM is sufficient for the given set of pseudo random numbers generated from three different sources.

E. Multiclass SVM

Multiclass SVM aims to label the test datasets of random numbers by exploiting SVMs. In this, the main approach used is to convert the multiclass problem into binary problem which will be done multiple times.

There are some methods through which such conversion is possible. Such methods include: (i) one-versus-one (or Pairwise approach), (ii) one-against-all. In the Pairwise classification, every classifier is assigned with instance to one of the classes. Then, the votes are counted for the assigned class. Finally, the class with maximum number of votes is assigned the instance classification.

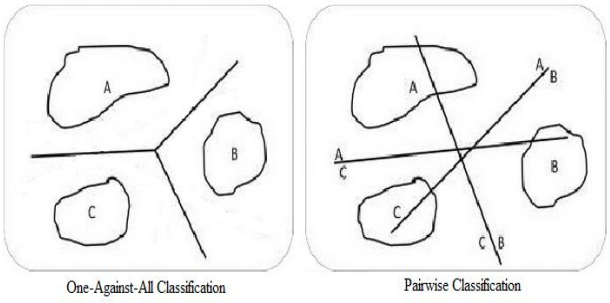

Fig. 3. Multiclass SVM

For this research work, one-versus-all approach is used. In this approach, Tested dataset is classified using an output function whose highest value defines the class to which that data is to be allocated.

III. EXPERIMENTAL VIEW

Here, multiple 1-dimensional data points to be categorised into three different classes. These different types of classes are the Pseudo Random numbers generated using three different types of PRNGs. Thus we have vectors. In addition to binary classification, we also scrutinize the use of multiclass classification for the problem at hand which we have used in this research. In one-against-all classification approach, there is a single binary SVM is exploited for each class so that members of that class can be separated from the members of other classes. We create sequences of Pseudo Random numbers and extract attributes pertaining to runs of various lengths which are then normalized and given as input to the SVM for training and cross validation. These three types of Pseudo Random numbers are first mixed and used as a single collection of all the numbers. A different set of sequences is generated by changing the bias of occurrence of particular outcomes and this set is tested on the trained classifier. And the algorithm for the same process is shown here as under:

Algorithm

Classification of Pseudo random numbers generated from three different types of sources using SVM Classifier.

Input: I: Input data, V: Support vectors

1. Assigning different class labels to each of the three set of pseudo random numbers, divide them into their

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

classes for training.
2. Add these training datasets to the support vector set named as V.
3. Then make a loop for these divided set of random numbers.
4. If any random number does not belongs to any of the class labels, then add that random number to the set V.
5. Break, if inadequate random numbers are found.
6. end loop.
7. Trained exploiting the resulting classifier i.e. SVM model.
8. Test, with the unlabelled random numbers, so as to validate the results.

IV. RESULTS

Now, in the results of the research, matlab simulation is shown. Matlab simulation shows the results as the outcomes of classification of the Pseudo Random numbers using the SVM classifier. In the fig. 4 shown, three classes (Class 1, 2 and 3) of Pseudo Random numbers are classified using SVM classifier.

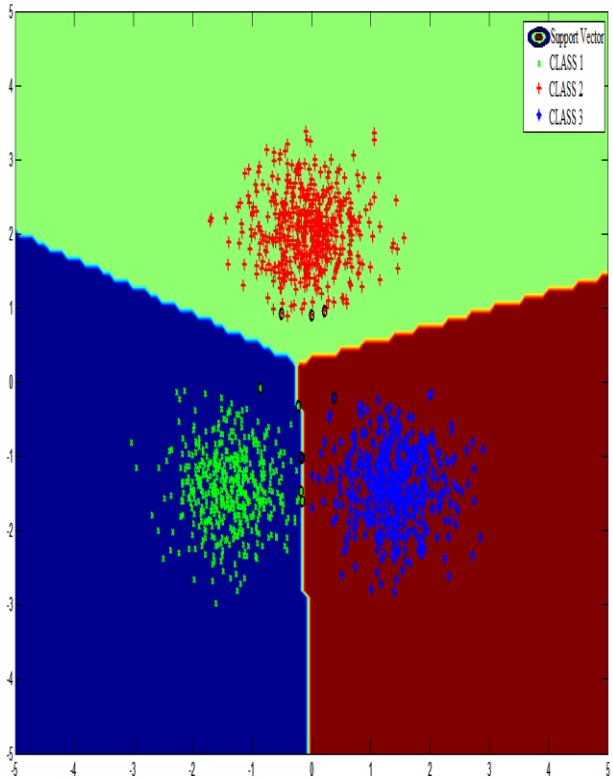

Fig. 4. Classification using SVM

As it was already shown in the fig. 1 that, firstly, three types of Pseudo Random nubmers generated using three different sources were mixed. Then these are very well classified by the SVM classifier using the algorithm that is already discussed in section III. Random numbers shown with green colour is class 1, red colour is class 2 and shown with blue colour is class 3. From the results, it can be seen that Pseudo Random numbers are very well separated using the SVM classifier. And it shows nearly 100% accuracy in classifying the Pseudo Random numbers.

V. CONCLUSION

This paper shows the successful classification of Pseudo Random numbers in their one-to-one classes using Support vector machine (SVM) classifier. We are able to get the optimum path to get the specific results which other classifiers were not able to deliver accurately and this accuracy can be further improved by using some more invariances as the features which can be used to classify these Pseudo Random numbers.

Our research can be helpful in the field of cryptanalysis. Cryptanalysis includes security in sending a message. Hence, this classification of Pseudo Random number's sources helps in enhancing that level of security in sending those messages. This enhancement in cryptanalysis surely booms the field of security.

REFERENCES

[1] J. K. Heinrich, Pangrat and Hans Weinrichter, "Pseudo Random Number Generator Based on Binary and Quinary Maximal-Length Sequences", Ieee Transactions On Computers, Vol. C-28, No. 9, September 1979

[2] D. H.Lehmer, "Random number generation on the BRL highspeed computing machines," by M. L. Juncosa. Math. Rev. 15, 559, 1954.

[3] L. Blum, M. Blum, and M. Shub. A simple unpredictable Pseudo Random number generator. SIAM Journal on computing, 15(2):364–383, 1986.

[4] V. N. Vapnik, "An overview of statistical learning theory", IEEE Transactions on Neural Networks, Volume: 10, Issue: 5, Pages: 988 – 999, 1999.

[5] F. Dehghan; H. Abrishami-Moghaddam, "Comparison of SVM and neural network classifiers in automatic detection of clustered microcalcifications in digitized mammograms", 2008 International Conference on Machine Learning and Cybernetics, Volume: 2, Pages: 756 – 761, 2008.

[6] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. Technical Report 23, Universit~it Dortmund, LS VIII, 1997.

[7] Koby Crammer and Yoram Singer, "On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines", Journal of Machine Learning Research 2 Page: 265-292, 2001.

[8] R.C. Fairfield, R.L. Mortenson, and K.B. Coulthart. An LSI Random Number Generator (RNG). In Advances in Cryptanalysis: Proceedings of Crypto 84, pages 203–230. LNCS 0196, Springer-Verlag, 1984.

[9] D. Eastlake, S. Crocker, and J. Schiller. Randomness recommendations for security. Network Working Group, RFC 1750, 1994.

[10] U. Vazirani and V. Vazirani. Efficient and secure Pseudo Random number generation. In Proc. 25th IEEE Symp. on the Foundations of Comput. Sci., pages 458–463, 1984.

[11] T. Stojanovski, J. Pil, and L. Kocarev. Chaos-based random number generators. Part II: practical realization. IEEE Transactions on Circuits and Systems – I: fundamental Theory and Application, 48(3):382–385, March 2001.

[12] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. Handbook of Applied Cryptanalysis. CRC Press, 5th edition, 2001.

[13] R.C. Fairfield, R.L. Mortenson, and K.B. Coulthart. An LSI Random Number Generator (RNG). In Advances in Cryptanalysis: Proceedings of Crypto 84, pages 203–230. LNCS 0196, Springer-Verlag, 1984.

[14] A New Approach to Machine Generation of Random Variables With Any Distribution P. T. Jeswani S. Sikdar, Senior Member Ieee Ieee Transactions On Reliability, Vol. R-27, No. 1, April 1978.

[15] C. Q. Li, S. J. Li, and G. Alvarez. Cryptanalysis of two chaotic encryption schemes based on circular bit shift and xor operations. Physics Letters (a), 369:23–30, 2007.

[16] C. Cortes and V. Vapnik. Support vector networks. Machine Learning, 20:273–297, 1995. R. Courant and D. Hilbert. Methods of Mathematical Physics. Interscience, 1953.

[17] C.J.C. Burges. Geometry and invariance in kernel based methods. In B. Sch¨olkopf, C.J.C. Burges, and A.J. Smola, editors, Advances in Kernel Methods: Support Vector Learning, pages 89–116. MIT Press, 1999.