International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:2, 2007

Applications of Cascade Correlation Neural Networks for Cipher System Identification

B. Chandra, and P. Paul Varghese

***Abstract*—**Crypto System Identification is one of the challenging tasks in Crypt analysis. The paper discusses the possibility of employing Neural Networks for identification of Cipher Systems from cipher texts. Cascade Correlation Neural Network and Back Propagation Network have been employed for identification of Cipher Systems. Very large collection of cipher texts were generated using a Block Cipher (Enhanced RC6) and a Stream Cipher (SEAL). Promising results were obtained in terms of accuracy using both the Neural Network models but it was observed that the Cascade Correlation Neural Network Model performed better compared to Back Propagation Network.

***Keywords*—**Back Propagation Neural Networks, Cascade Correlation Neural Network, Crypto systems, Block Cipher, Stream Cipher.

I. Introduction

INFORMATION security is one of the major concerns in cyber age. Cryptology is extremely vital for information security. It has wide applications in defence. Today the advances in cryptology favor the designer of cryptosystem in the sense that one can implement a family of complex algorithms with a very large key domain posing a great challenge to the analyst. The crypt analyst will generate reasonably sufficient amount of cipher texts for capturing the statistically significant signature of systems. For any new cipher text the same parameter will be computed and matched with the already learned systems and if the new cipher belongs to one of the already learned systems, then the system is identified for the new ciphers. System Identification helps in reducing the effort of crypt analyst to a considerable extent.

Cryptanalysis plays an important role in the development of strong cryptographic algorithms. It helps researchers in identifying the vulnerability of the cipher system. With advancement in computing technology and mathematical sciences, cryptographic algorithms have to be made more sophisticated so that the computational complexity and the time required to break into the cipher systems is too large. Until now various techniques have been suggested by cryptanalysts. However, it is seen that machine leaning techniques have not been used in cryptanalysis successfully. Machine learning techniques are capable of capturing the underlying variation in the data and making effective use of them for classification and prediction. Neural Networks have found its applications in various fields of science and are effective machine learning tool. The advantage of Neural Networks is that it is a powerful data-modeling tool, which has the ability to recognize patterns even if there is no functional relationship between input and output. We have tried to explore the possibility of the use of neural networks for identification of crypto systems.

The most commonly used Neural Network model is the Back-propagation Network [3] which is a multi-layer feed-forward network. The disadvantage with multi-layer feed-forward networks using error back propagation is that the best number of hidden layers and units varies from task to task and so must be determined experimentally. If too many hidden units are used then the network will learn irrelevant details in the training set and once trained will not generalize well. Conversely, if a network is too small, it will be unable to learn the training set properly. One approach to automatically determine a good size for a network is to start with a minimal network and then add hidden units and connections as required like the Cascade Correlation Neural Networks. Cascade Correlation Neural Networks [1] helps in overcoming this shortcoming by increasing the number of hidden layers dynamically during the learning phase. Crypto system identification between block cipher Enhanced RC6 [2] and stream cipher SEAL [4] has been attempted using Cascade Correlation and Multilayered Feed forward Neural Network. Large number of cipher texts was generated using both cipher systems. It was observed that the Cascade Correlation Neural Network model performed better in comparison to Multilayered Feed forward Network using Back propagation.

B. Chandra and P. Paul Varghese are with the Indian Institute of Technology, Delhi, India (corresponding author is B. Chandra, phone: 91-11-26591998; fax: 91-11-25581005; e-mail: bchandra104@yahoo.co.in, pallathpv@yahoo.com).

II. Description of Cipher Systems used for Identification

The description of the block and stream cipher systems used for identification are given below

A. Enhanced RC6

Enhanced RC6 [2] is a block cipher with block size of 256bits. It uses an expanded key table S that is derived from the user supplied secret key. The size of S table is dependent on the number of rounds. Size of S table is 4r+8 words where

r denotes the number of rounds. Choosing a larger number of rounds 'r' may provide an increased level of security. It uses a variable length cryptographic key, which could be a maximum of 255 characters.

B. Seal

The Stream Cipher SEAL (Software Encryption Algorithm) [4] is a pseudorandom function family: under control of a key, first preprocessed into a set of tables, SEAL stretches a 32 – bit "position index" into a key stream of essentially arbitrary length. One then encrypts by XORing this keystream with the plaintext, in the manner of Vernam Cipher. In a stream cipher the encryption of a message depends on Key 'a', the message x, and the messages 'position' n in the data stream. This position is often a counter (sequence number 0 which indicates which message is being enciphered. The encryption of string x at position n is given by <n, x ⊕ SEAL(a,n,L) > ,where L= | x |.

III. Neural Network Algorithms

Overview of various Neural Network models used of Crypto System Identification is given in this section.

A. Back Propagation Algorithm

Enhanced The standard Back propagation algorithm [3] can train any network as long as its weights, net input and transfer functions have derivative functions. Here the weights are adjusted according to gradient descent.

$$\Delta w = -k\frac{\partial E}{\partial w} \tag{1}$$

$$= \eta\frac{\partial E}{\partial w} \tag{2}$$

where η is the learning rate , Δw is the weight change and E is the sum of squares of error

The problem with the standard gradient descent method is that it at times gets trapped into local minima, and hence variations were suggested. In Gradient descent algorithm with momentum, the weights are adjusted according to gradient descent. However some weightage is given to the previous weight change also.

$$\Delta w_n = \alpha\Delta w_{n-1} + \eta\left(1-\alpha\right)\frac{\partial E}{\partial w} \tag{3}$$

where α is the smoothing factor for applying the momentum and η is the learning rate.

B. Cascade Correlation Neural Networks

Cascade-Correlation (CC) [1] is an algorithm developed by Scott Fahlman. The procedure begins with a minimal network that has some inputs and one or more output nodes as indicated by input/output considerations, but no hidden nodes. The gradient descent algorithm, for example, may be used to train the network. The hidden neurons are added to the network one by one, thereby obtaining a multilayer structure. Each new hidden neuron receives a synaptic connection from each of the input nodes and also from each preexisting hidden neuron. When a new hidden neuron is added, the synaptic weights on the input side of that neuron are frozen; only the synaptic weights on the output side are trained repeatedly. Learning in CC takes place as repeating two phase steps. The first involves the embedded standard learning algorithm, which in our case is Back propagation (BP). The learning algorithm begins with no hidden units. The direct input-output connections are trained as far as possible over the entire training set. When no significant error reduction has occurred after a certain number of training cycles (controlled maximum number of epochs set by the user), the entire training set is presented to the network one last time to measure the error. If the network's performance is satisfactory, we stop; if not, there must be some residual error that needs to be reduced further. This is achieved by adding a new hidden unit to the network, using the unit-creation algorithm.

The second phase involves the creation of a 'candidate' unit. The candidate unit is connected with all input units and all existing hidden units. It is this which leads to the cascading architecture, as each new unit is connected to all preceding units. There are no connections from these candidate units to the output units. The links leading to each candidate unit are trained with the selected standard learning algorithm (BP) to maximize the correlation between the residual error of the network and the activation of the candidate units. The goal of this adjustment is to maximize *S*, the sum over all output units *o* of the magnitude of the correlation1 between *V*, the candidate unit's value, and *Eo*, the residual output error observed at unit *o*. *S* is defined as

$$S = \sum_o \left| \sum_p \left(V_p - \overline{V}\right)\left(E_{p,o} - \overline{E_o}\right) \right| \tag{4}$$

where o is the network output at which the error is measured and p is the training pattern. The quantities $\overline{V}$ and $\overline{E_o}$ are the values of V and E_o averaged over all patterns.

In order to maximize *S*, $\partial S / \partial w_i$, the partial derivative of *S* with respect to each of the candidate unit's incoming weights, w_i is computed.

$$\partial S \Big/ \partial w_i = \sum \sigma_o \left(E_{p,o} - \overline{E_o}\right) f_p' \; I_{i,p} \tag{5}$$

where σ_o is the sign of the correlation between the candidate's value and output o, f_p' is the derivative for pattern p of the candidate unit's activation function with respect to the sum of its inputs, and $I_{i,p}$ is the input the candidate unit receives from unit i for pattern p.

After computing $\partial S/\partial w_i$ for each incoming connection, gradient ascent is performed to maximize *S*. Once again only a single layer of weights are trained. When *S* stops improving, we install the new candidate as a unit in the active network, freeze its input weights, and continue the cycle as described above. Because of the absolute value in the formula for *S*, a candidate unit cares only about the *magnitude* of its

correlation with the error at a given output, and not about the sign of the correlation. As a rule, if a hidden unit correlates positively with the error at a given unit, it will develop a negative connection weight to that unit, attempting to cancel some of the error; if the correlation is negative, the output weight will be positive. Since a unit's weights to different outputs may be of mixed sign, a unit can sometimes serve two purposes by developing a positive correlation with the error at one output and a negative correlation with the error at another.

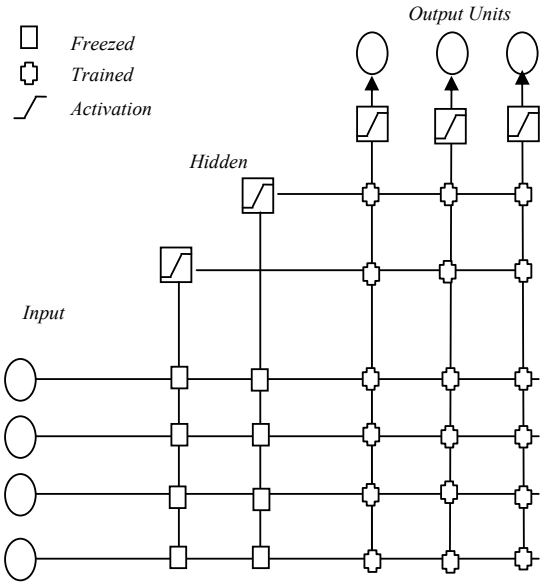

Fig. 1 Cascade Correlation Neural Network

Training is stopped if the correlation ceases to improve or a predefined number of cycles are exceeded. The final step of the second phase is the inclusion, as a hidden unit, of the candidate unit. This involves freezing all incoming weights (no further modifications will be made) and creating randomly initialized connections from the selected unit to the output units. This new hidden unit represents, as a consequence of its frozen input connections, a permanent feature detector. The weights from this new unit and the output units will undergo training. Because the outgoing connections of this new unit are subject to modification its relevance to the final behavior of the trained network is not fixed. These two phases are repeated until either the training pattern has been learned to a predefined level of acceptance or a preset maximum number of hidden units have been added, whichever occurs first.

The performance of the Cascade Correlation Neural Network model is compared to that of Back Propagation Neural Network model after applying suitable transformations to the ASCII representations of the Cipher texts. The following section discusses results of classification of crypto system using Neural Network.

IV. Results

Large collection of cipher texts was generated from different sets of plaintext messages (12.8K Bytes) using Enhanced RC6 and SEAL. For crypto system identification, the ASCII values of the cipher texts were considered. A group of 32 ASCII values were considered as one input pattern. The message of 12.8K Bytes resulted in 400 patterns each containing 32 ASCII values of the corresponding Cipher characters. Various Neural Network models were trained using this data but the testing accuracy was low. This is due to the fact that the Cipher text characters are a pseudorandom sequence. Various standard data transformation techniques like 1/x, $\sqrt{x}$, log x was applied on the cipher text data for improving the testing accuracy but the results were not encouraging. Summation of 200 patterns was taken and then appropriate transformations were carried out to make the series stationary. The resultant patterns were fed then fed to the Neural Network.

A. Data Set 1: 0.98 Million Cipher Characters, Single Key, Different Plaintext Messages

Dataset 1 compromises of the collection of cipher texts generated using different plaintext messages encrypted using both the encryption algorithms. The Key used for both the Cipher systems was the same. 0.98 Million Cipher Characters were transformed using the data transformation technique and 0.7 Million Cipher Characters were used for training and 0.3 Million Cipher Characters were used for testing. The performance of Cascade Correlation Neural Network and Back Propagation Neural Network are shown in Table I. The results stated in the table are the best results achieved after tuning various neural network parameters.

The Cascade Correlation Neural Network was trained for the Dataset 1, for different values of the Goal parameter, learning rate and the momentum factor. The Best Results were achieved when the Network was trained using Learning rate =0.3 and Goal =0.0005. The Classification Accuracy achieved was 90.9% with 2 hidden layers dynamically added. It was also observed that with the reduction of Goal Parameter the accuracy also came down. Back Propagation Neural Network gave the best classification accuracy of 73% for a two hidden layer network (10 neurons in first hidden layer and 3 neurons in the second hidden layer) when trained with learning rate of 0.5 and 0.005 goal.

TABLE I
Results For Dataset 1(Single Key, Different Messages)

Neural Network Model	Accuracy	HL	GL	η	O_Epoch	C_Epoch
Cascade Correlation	90.9 %	2	0.0005	0.3	1000	1000
Gradient descent back propagation	73 %	2	0.005	0.5	Convergence reached in 476 Epochs	

HL=No of Hidden Layers; GL = Goal; η = Learning Rate; O_Epoch = Max Epoch for Output Layer; C_Epoch = Max Epochs for Candidate Layers.

B. Data set 2: 0.96 Million Cipher Characters, Same Set of Plaintext messages, Different sets of Keys

Dataset 2 compromises of the collection of cipher texts generated using same set of plaintext messages encrypted using different sets of Keys for both the encryption algorithms. Out of the 0.96 Million Cipher characters generated after the data transformation technique, 0.74 Million Cipher Characters were used for Training and 0.26 Million Cipher Characters were used for testing. The results of Cascade Correlation Neural Network and Gradient Descent Back Propagation Neural Network for dataset 2 are shown in Table II. The Neural Network architectures were kept as discussed for dataset1.

TABLE II
RESULTS FOR DATASET 2 (DIFFERENT KEYS, SAME MESSAGES)

Neural Network Model	Accuracy	HL	GL	η	O_Epoch	C_Epoch
Cascade Correlation	93 %	2	0.0005	0.3	1000	1000
Gradient descent back propagation	83 %	2	0.005	0.5	Convergence reached in 556 Epochs	

HL=No of Hidden Layers; GL = Goal; η = Learning Rate; O_Epoch = Max Epoch for Output Layer; C_Epoch = Max Epochs for Candidate Layers.

Cascade Correlation Neural Network has given the best results of 93 %, also the only 2 hidden layers dynamically added. However the Back Propagation Neural Network gave 83% accuracy.

C. Data Set 3: 1.92 Million Cipher Characters, Different Sets of Plaintext Messages, Different sets of Keys

Dataset 3 compromises of the collection of cipher texts generated using different plaintext messages encrypted using different sets of Keys for both the encryption algorithms. After data transformation, 1.66 Million Cipher Characters were used for training and 0.26 Million Cipher Characters were used for testing. The Performance of both the Neural Network models in identifying Cipher systems for dataset 3 is shown in the Table III.

The Classification Accuracy of Cascade Correlation Neural Network is 92 %. Here classification accuracy of RC6 was 94% and that of SEAL was 90 %. Two Hidden Layers were added dynamically in the Cascade Correlation Neural Network.

V. CONCLUSION

Observing the results of dataset1, dataset2 and dataset3 it can be concluded that even with the increased complexity of the datasets, the results have been consistently well and also that the Cascade Correlation Neural Network Model out performs gradient descent Back Propagation Neural Network Model. It has been successfully possible to train the neural network models to classify cryptosystems.

TABLE III
RESULTS FOR DATASET 3 (DIFFERENT KEYS, DIFFERENT MESSAGES)

Neural Network Model	Accuracy	HL	GL	η	O_Epoch	C_Epoch
Cascade Correlation	92 %	2	0.0005	0.3	1000	1000
Gradient descent back propagation	85 %	2	0.005	0.5	Convergence reached in 672 Epochs	

HL=No of Hidden Layers; GL = Goal; η = Learning Rate; O_Epoch = Max Epoch for Output Layer; C_Epoch = Max Epochs for Candidate Layers.

REFERENCES

[1] S. E. Fahlman, and C. Lebiere, "The cascade-correlation learning architecture", in Advances in Neural Information Processing Systems 2, San Mateo, CA: Morgan Kaufmann, 1990,pp. 524-532.

[2] A. H. M. Ragab; N. A. Ismail; O. S. F. Allah.; "Enhancements and implementation of RC6TM block cipher for data security", Electrical and Electronic Technology, 2001. TENCON. Proceedings of IEEE Region 10 International Conference, 19-22 Aug.2001, vol. 1, pp. 133 – 137.

[3] D. E. Rumelhart; G. E. Hinton and R. J. Williams; "Learning internal representations by error propagation", Parallel Data Processing, Vol.1, Chapter 8, the M.I.T. Press, Cambridge, MA, 1986, pp. 318-362.

[4] P. Rogaway and D. Coppersmith., "A software-optimized encryption algorithm", Journal of Cryptology, 1998, vol. 11, num 4, pp. 273-287.