Validation Testing for Temporal Neural Networks for RBF Recognition

Khaled E. A. Negm

Abstract—A neuron can emit spikes in an irregular time basis and by averaging over a certain time window one would ignore a lot of information. It is known that in the context of fast information processing there is no sufficient time to sample an average firing rate of the spiking neurons. The present work shows that the spiking neurons are capable of computing the radial basis functions by storing the relevant information in the neurons' delays. One of the fundamental findings of the this research also is that when using overlapping receptive fields to encode the data patterns it increases the network's clustering capacity. The clustering algorithm that is discussed here is interesting from computer science and neuroscience point of view as well as from a perspective.

Keywords—Temporal Neurons, RBF Recognition, Perturbation, On Line Recognition.

I. INTRODUCTION

RTIFICIAL neural networks (ANNs) whose functioning Ais inspired by some fundamental principles of real biological neural networks have proven to be a powerful computing paradigm. Real biological neurons communicate through short pulses, called spikes, which terminate at different time rates. While firing, the firing rate is considered as the relevant information exchanged occasion between neurons, where the analog inputs for an artificial neuron are interpreted as firing rates. A spiking neuron is a simplified model of the biological neuron, however it is more realistic than a threshold gate (perceptron) or a sigmoidal gate. One reason for this is that in a network of spiking neurons the input, output and internal representation of information is more closely related to that of a biological network. This representation allows for time to be used as a computational resource and enhance neural network performance. It has been shown that such a network is computationally more powerful than a network of threshold or sigmoid gates. Because of this high efficiency many researchers have recommended to use such techniques not only for off-line real data classification but also for on-line high rate data classification. However; learning algorithms for spiking neural networks are still not adequate to serve this purpose

[1,2].

In order to understand the neural encode we have to investigate the temporal structure of the spiking neurons [2-4], where neurobiological findings have confirmed such dependency and have shown that the sign and strength of the change depends on the timing of the multi-spike systems [3-7].

The importance of the timing of the first spike has been discussed by many authors in which they were able to show that humans can process visual patterns in 150 ms [8]. Within this time it is hard to imagine that the neurons may sample firing rates, since there are only about 10 synaptic stages involved. Neurons participating in such computations usually have a firing rate of less than 100 Hz and hence 10 *ms*, are not sufficient to estimate the current firing rate of some spiking neuron [9].

So far there is no much information known about the possible computational mechanisms on the basis of the timing of single spikes. Some fundamental results have been provided by Maass in which he characterized the computational power of SNNs and showed that the timing of spikes can be used to simulate sigmoidal gates with SNNs [10,11]. Other studies, by Allipi et al. investigated the correlation between the accuracy of the system versus its complexity and its influence on off-line and on-line data analysis [12].

On the basis of these principles, in the present study we show how methods originally designed for artificial neural networks like competitive learning, self-organizing behavior and radial basis functions (RBF) can be realized. Within this context we investigated the effect of using inner time dependence for firing which shows promising results in terms of real time (on line) execution time for recognition and less resources for that training sets.

II. CLASSIFICATION OF NEURON'S ENCODES

One can classify the neural information encoding by three different approaches: the first is the *rate coding* where the essential information is encoded in the firing rates and averaged over time or over several repetitions of the event. The *second*, is the *population coding*, where information can be distinguished by the activity of different populations of neurons where a neuron may participate at several pools. And

Manuscript received March 19, 2005. (Write the date on which you submitted your paper for review.)

K. E. Negm is with the Etisalat College of Engineering, Sharjah, POB 980, UAE (corresponding author to provide phone: 50-482-1316; fax: 6-522-5937; e-mail: knegm@eim.ae).

the *third*, is the *temporal coding*, where the timing of single spikes is used to encode information.

In the present study we will focus on the temporal coding in which relevant information could be represented. In this context we consider that the firing rates of neurons are relative to the stimulus onset such that the closer a neuron fires to the onset the stronger the stimulation can take effect [13]. In previous studies, researchers considered only the first spike of a neuron carries relevant information in which the neuron is shut off by some additional inhibitory input after its firing [14,15].

In the current study we introduce new novel scheme that is composed of two new techniques; the *first* is that all the neurons are participating in the firing but according to a normalized time dependent correlation function. In that sense the overall firing behavior is a superposition of the firing rates from the neurons which is more elastic to real life. The *second*, is that the firing rates in total is calculated from all neurons participating in the event an weighted as perturbation terms, in which the order of perturbation depends on the location of the firing neuron from the principle neuron, i.e., the most principle participating one. The perturbation factor is relative to the spatial distance of the designated neuron from the principal one.

A. Modified Model for Temporal Neurons

The state of neuron *j* is described by the variable $x_j(t)$, which models the neuron's membrane potential (Excitatory Post Synaptic Potential-ESPS) at time *t*. A spike is generated whenever $x_j(t)$ crosses the threshold and $\Gamma_j = \{t_j^1, t_j^2, t_j^3, ...\}$ is the set of firing times of neuron *j*. We can model the effect of an incoming spike on the ESPS by in which the first term is the normalized Gaussian Probability Integral

$$\varepsilon(t) = \begin{cases} \left(2\sqrt{\frac{\delta}{\pi}}\int_{0}^{\infty} \exp(-\delta t^{2})dt\right) \left(\frac{\delta}{\tau}\exp(1-\frac{\delta}{\tau})\right), & \forall \ \delta > 0 \\ 0, & \forall \ \delta < 0 \end{cases}$$
(1.a)

where δ is the post time to the spike effect and τ is the time to the peak of the post synaptic reaction. Also can argue that:

$$\lim_{\sigma,\tau \to very \ small} 2\sqrt{\frac{\delta}{\pi}} \int_{0}^{\infty} \exp(-\delta t^{2}) dt = 2\sqrt{\frac{\delta}{\pi}} \int_{0}^{\tau} \exp(-\delta t^{2}) dt, \ (1.b)$$

Different approaches have used the second bracket only of the right hand side, which is not physically correct for the potential function normalization. It is obvious that equation (1.a) is more physically acceptable because it represents a normalized potential function.

Assuming that the input is of a vector form and defined as: $\mathbf{x} = \langle x_1, \dots, x_m \rangle$ (2)

where

$$x_{i} = \max\{t_{i} \mid 1 \le j \le m\} - t_{i}(3).$$

When the neuron *j* receives an input from a set of neurons marked as Γ_j , the membrane potential becomes the weighted sum of EPSP's caused by the neurons from Γ_j . In this case

$$x_{j}(t) = \sum_{j \in \Gamma_{i}} \sum_{t_{j}^{k} \in \Gamma_{j}} w_{ij} \varepsilon(t - (t_{j}^{k} + d_{ij}^{k}))$$
⁽⁴⁾

where w_{ij} is the synaptic efficiency (i.e., weights) between neurons *i* and *j*, d_{ij}^k is the delay from the occurrence of a spike in neuron *j* and the beginning of its effect on neuron *i* and $(t_j^k + d_{ij}^k)$ is the time when the k^{th} spike from neuron *j* started affecting neuron *i*.

B. RBF and Temporal Data Encoding

Radial Basis Functions (RBF) represents an approach for the universal function approximation. RBFs were first used in solving multivariate interpolation problems and numerical analysis. Its prospect is similar in neural network applications, where the training and query targets are rather continuous. RBF network performs a local mapping (i.e., only inputs near specific receptive fields will produce an activation)., where, the units (in the hidden layer) receiving the direct input from a signal may see only a portion of the input pattern, which is further used in reconstructing a surface in a multidimensional space that furnishes the best fit to the training data.

RBF models the response function using the composition of sigmoid-cliff functions - for a classification problem; this corresponds to dividing the pattern space up using circles or (more generally) hyperspheres. A hypersphere is characterized by its center and radius. More generally, just as an RBF unit responds (non-linearly) to the distance of points from the line of the sigmoid-cliff, in a radial basis function network units respond (non-linearly) to the distance of points from the center represented by the radial unit [16-18]. The response surface of a single radial unit is therefore a Gaussian function, peaked at the center, and descending outwards in which the sigmoid curves can be altered, so can the slope of the radial unit's Gaussian. See the next illustration below.

RBF, therefore, has a hidden layer of radial units, each actually modeling a Gaussian response surface. Since these functions are nonlinear, it is not actually necessary to have more than one hidden layer to model any shape of function in which sufficient radial units will always be enough to model any function. The remaining question is how to combine the hidden radial unit outputs into the network outputs? It turns out to be quite sufficient to use a linear combination of these outputs (i.e., a weighted sum of the Gaussians) to model any nonlinear function. The standard RBF has an output layer containing dot product units with identity activation function [18,19]. In the present study we used instead a nonlinear combination weighted by perturbation terms related to the distance from the principle firing neuron.

Assume that *m*-dimensional input patterns $(x_1,...,x_m)$ are encoded with sensory neurons $v_1,...,v_m$ which are the constituents of the networks' input that have an *n*-dimensional output neurons $(u_1,...,u_n)$ (see Figure a). In the simplest case each input neuron v_i connections to the RBF neuron u_j with weights w_{ij} and delays d_{ii}^k . In the present study we consider the encoding scheme where each input neuron u_i fires exactly once at time t_i during the encoding interval $[t_i, (\delta_t + t_i)]$, i.e., with δ_t is a constant time step, which is encoded relative to the first spike in the coding interval.



Figure a. The SNN encoding architecture

We can define the center of an RBF neuron (which is symmetric around the centre) c_j by the vector $x_i = \langle x_i, ..., x_{i,m} \rangle$ (5)

where

$$x_{i,m} = d_{im} - \min\{d_{im} \mid 1 \le i \le m\}$$
(6)

Assuming that the RBF neuron v_j is associated with a *m*-dimensional vector as in equation (5). In that sum we define the *m*-dimensional vector as the synaptic weights of the neuron. Each coordinate in μ_j^c is a weighted average of the delays from the matching input coordinate as in equation (4). The active synapse from μ_j will be the one whose delay defined as

$$w_{ij}^{d} = \begin{cases} w_{\max}, & d = \mu_{j}^{i} \\ 0, & \text{otherwise} \end{cases}$$
(7)

The center of such a neuron is mirror image of μ_j^p and the response time to input patterns is symmetric around that center so that this neuron realizes a RBF.

For each pair of sensory neuron u_i that is connected to a single layer of *n* spiking neurons and RBF neuron v_j , there exist a set of *D* independent synapses. Taking into account the multiple synapses and constant delays, the weights of these synapses are $w_{ij}^1, w_{ij}^2, ..., w_{ij}^D$ and the delays are 1 *ms*, 2 *ms*, ... and *D ms*, respectively. In this case $x_j(t)$ is redefined as

$$x_{j}(t) = \sum_{j \in \Gamma_{j}} \sum_{t_{j}^{k} \in \Gamma_{j}} \sum_{d=1}^{D} w_{ij}^{d} \varepsilon(t - (t_{j}^{k} + d_{ij}^{k}))$$
(8)

The input \mathbf{x} is close to the center c_j of an RBF neuron v_j if the spikes of the input neurons reach the soma of v_j due to the corresponding delays at similar times, i.e., if $\|\mathbf{x}-\mathbf{c}_j\|$ is small. This is basically a parallel approach to that was introduced by Hopfield in 1995 in which he considered the case where the input vector is close enough to the center of an RBF neuron to make v_j fire [20].



If the distance between x and v_j is too large, v_j does not fire at all. If for some input vector x the difference $||x-c_j||$ is small enough for various j to make v_j to fire then the RBF neuron whose center is closest to x fires first. In this case a set of such RBF neurons can be used to separate inputs into various clusters.

Figure b, shows the dependence of the firing time of an RBF neuron v_j on the distance of the input vector \mathbf{x} to the center c_j . For this simulation 1200 uniformly distributed inputs $\mathbf{x} \in [0, 30ms]^{1200}$ are presented to v_j with equal weights and delays uniformly distributed over [0, 30ms]. Crosses indicate the case that the RBF neuron has not fired [21].

The delays between spikes in such an input pattern will be evened out by the delays in the synapses causing v_j to react to the all incoming spikes at the same time. When comparing the reaction of v_j to an input pattern which is at its center and an input pattern which is slightly off its center we find that the first pattern causes a higher peak in v_j 's membrane potential and the membrane potential crosses the threshold earlier.

C. Temporal Neurons and RBF Recognition

In Figure c, the solid line is the response to a pattern that is directly at the center of the RBF, the dashed line is the response to an input slightly off the center and the other two patterns are farther away. If the threshold is set to 10, the neuron will react fastest to the pattern which is at the center. When presented with the pattern farthest from the center (dashed-dotted line) it will not reach the threshold, and therefore not fired at all. The farther away the input pattern is from the center the later the neuron will reach the threshold. When a pattern is very different from the center the peak will be lower then the threshold and no spike will be generated. The neuron realizes a function whose input is an *m*-

dimensional pattern and output is the time till it emits a spikeif at all-this function is symmetric around the center defined and therefore it is a radial basis function.



Figure c: Membrane potential as a result of four 15-dimensional input patterns

D. Learning RBF and Clustering with Temporal Neurons

The learning goal of the network is to have one RBF neuron related to each cluster so that when an input pattern from that cluster is exposed to the network, only the related neuron will fire. Since we have inhibitory synapses between the neurons, it is enough that the correct neuron will fire first. In order to achieve that, the weights of the spiking neurons will be shifted during learning so that they will be able to realize RBFs whose centers are the centers of the clusters.

The learning rule (which is a variant of the Hebb law) is applied to the synaptic weights of the neuron that is fired when the input is fed to the network. Synapses which contributed to the neuron's firing are strengthened and synapses which did not contribute are weakened by perturbation. The synapses that contributed are those who started affecting the postsynaptic neuron (taking into account the synaptic delay) slightly before the neuron actually crossed the threshold. The change in the synaptic weight is given by the following learning rule

$$\Delta w_{ii}^{d} = \varepsilon \left(t_{i} - \left(t_{i}^{k} - d_{i}^{k} \right) \right)$$
(9)

where ε determines the learning rate. The learning function will have the form

$$\alpha(\Delta t) = \varepsilon (1-b) \exp\left(-\frac{(\Delta t - c)^2}{\beta^2} + b\right) (10)$$

where b is the minimal value, c is the location of the peak, β is the width of the distribution.

We found out that with time difference of exactly c ms between the spikes causes maximal strengthening of the synaptic weight up to the width of the distribution (higher or lower) causes a smaller strengthening and weakens the synaptic weight. This is a very good result that guided us to the best resonance value for the system parameters. With the use of this above model data is successfully clustered with results similar to [22].

III. INPUT ENCODING WITH RECEPTIVE FIELDS

In order to encode the sensory input neurons we should look for an efficient encoding technique. Sensory input in live organisms are often encoded with overlapping receptive fields, for example touching the skin at a certain area may cause several sensory neurons to fire at different rates. This technique is used to encode input patterns so that it is possible to successfully cluster more complicated data sets.

When using receptive fields, input is not encoded by using just one sensory neuron for each data coordinate. Instead for each coordinate, several receptive field neurons are used to encode the data. Each of the receptive fields fires with a short delay if the value of that coordinate is close to its center and with a longer delay for values farther from the center. And it does not fire at all if the value is too far from the center. The centers of the receptive fields are evenly distributed within the possible range of values.

When a pattern is introduced to the system each receptive field calculates the value of a gaussian function and value of the input at the coordinate will correlate with each other. The gaussian function for each receptive field is given by equation (10) when using

$$\beta = \beta_{rf} M/(y-2) (11)$$

where γ is the width of the receptive field, *M* is the maximal value of the input and *y* the number of receptive fields and $\beta_{\rm rf}$ a parameter.

IV. MODEL IMPLEMENTATION

The model was implemented by RBF++ using the three classes ReceptiveFiled, RBFneuron and Network Parameters' Monitoring [21]. The algorithm contains main three modules. The first one is data set file creation module, the second is data set file testing module and the third one is the distance from the centre, response time testing and running module.

A pre-requisite for the system is to define the values for parameters discussed before as:

- *Maximal weight value*, The maximal weight is calculated so that after learning the active synapses that are left have enough weight to cause the RBF neurons to fire. There is also a minimal weight which is set to zero in which no inhibitory synapses are allowed.
- Saturation function that is needed in order to keep the weights within the realistic range $[0, w_{max}]$. This function causes the synaptic weight to change at a slower rate when close to 0 or γ_{max} .
- *Initial weight value* where the weights are initiated randomly but they must be high enough to cause some neuron to fire for every pattern and low enough so that all the input spikes are necessary in order to cause a RBF neuron to fire. If not all input spike are necessary a partial pattern will be learned and several clusters with a common sub-pattern will be identified by one of the neurons.

An input (data-point) to the network is coded by a pattern of firing times within a coding interval ΔT and each input neuron is required to fire at most once during this coding interval. In our experiments, we set ΔT to [0-9] ms and delays

dk to 1 - 30 ms in 1 ms intervals (m = 31). For the reported simulations, the parameter values for the learning function are set to: b = 0.2.3, c = -2.70, $\beta = 1.2.64$, $\varepsilon = 0.0018$ and $w_{max} = 2.45$. To model the (strictly excitatory) post-synaptic potentials, we used an α -function:

$$\varepsilon(t) = \frac{t}{T} \exp\left[1 - \frac{t}{\tau}\right] (12)$$

where T is set to be 2.0 ms, effectively implementing leakyintegrate-and-fire spiking neurons.

V. RESULTS AND CONCLUSION

Throughout this work we incorporated receptive fields along with the distance from the center versus the firing time function of the RBF neurons under test. A single spiking neuron with one active synapse from each sensory neuron is stimulated with different input patterns and the time of firing is measured and compared to the distance of the patterns from the neuron's center. Figure d shows the results of this run, in which the spike time is measured relative to the minimal response time. Spike time "-1" indicates that the neuron did not fire as a result of that pattern.



Figure d: firing time of the RBF neurons versus distance of the input patterns from the RBF center.



The value of the Gaussian (between 0 and 1) is normalized to a value less than 1 and the result is multiplied by the maximal delay to the delay of the receptive field to that input. If the value of the Gaussian is too low no spike will be emitted by that receptive field. In Figure e, the vertical line indicates an input which causes three receptive fields to fire, the central receptive field will fire almost immediately and the ones to its left and right will fire after a longer delay of value 80% and 90% of the maximal delay.

In table I, shows the results for different data sets. We run the algorithm for different values of dimensions for the data sets. It shows very promising results in terms of the standard deviation relative to the number of data points.

TADLET

DATA SET PARAMETERS USED FOR NNS MODEL TESTING				
Dimension	# of	Range	Std	# of
	Clusters	ms	ms	patterns
2	20	0 - 30	0.2	2500
10	20	0 - 30	0.22	2500
20	20	0 - 30	0.23	3000
40	20	0 - 30	0.24	2000
60	20	0 - 30	0.26	3000
80	20	0 - 30	0.28	3000
4-Iris data	12			800



Figure f: Plot of the 2-dimensional data set results for the NNS testing.

In Figure f we shows the relation for the first row of the table, and it is obvious that it gives a very good results to discriminate and cluster data points in a very small response time.

In all the simulations 30% of the data was used for learning. Testing was performed on the full data sets. In the artificial data sets 100% correct classification was achieved. In the iris data set over 95% success was achieved. So we can see that spiking neurons, receiving temporally encoded inputs can compute radial basis function to an excellent accuracy. This is feasible via sorting the relevant information in their delays. In the current study we showed how our models introduced excellent results with simpler buildup than the previous studies.

FUTURE WORK

Currently we are studying applying this technique to more NNS application oriented problems. Of main interest to us to benefit from the short time convergence into correct clustering and very small standard deviation. Namely applying this for intrusion detection systems as extensions to our previous efforts in that field [22,23].

References

- W. Mass, Networks of spiking neurons: the third generation of neural network models, *Neural Networks*, 10, 1998. pp.1659-1671.
- [2] L. Guo, D-H. Huang, and W. Zhao, Combining Genetic Optimization with Hybrid Learning Algorithm for Radial Basis Function Neural Networks, Neural Networks, 2003. Proc. of the International Joint Conference on Neural Networks Vol. 4, 2003, pp.3213 – 3217.
- [3] T. S Sejnowski, Time for a new neural code?, *Nature*, 376, 1995, pp. 21-22., A
- [4] M. Abeles, M., A., Corticonics, Neural Circuits of the Cerebral Cortex, Cambridge University Press, 1991.,
- [5] M. Abeles, H. Bergman, H., Margalit, and E. Vaadia, Spationtemporal firing patterns in the frontal cortex of behaving monkeys, Journal of Neurophysiology, 70, 1993, pp. 1629-1638.
- [6] Z. Mainen, and T. Sejnowski, Reliability of spike timing in neocortical neurons, *Science*, 268, 1995, pp.1503-1506.
- [7] R. Rieke, F. Warland, D., Bialek, and R. Van Steveninck., SPIKES: Exploring the Neural Code, MIT-Press, Cambridge, 1996.
- [8] S. Thorpe, S. Fize, and C. Marlot, Speed of processing in the human visual system, *Nature*, 381, 1996, pp. 520-522.
- [9] S. Thorpe and M. Imbert, Biological constraints on connectionist modeling. In Pfeifer, R., Schreter, Z., Fogelman-Soulie, F., and Steels, L., editors, *Connectionism in Perspective*. Elsevier Publisher, 1989.
- [10] W. Mass, Lower bounds for the computational power of networks of spiking neurons. Neural Computation, 8, 1996, pp. 1-40, 1996.
- [11] W. Mass, Fast sigmoidal networks via spiking neurons, *Neural Computation*, 9, 1997, pp. 279-304.
- [12] C. Allipi, V. Piuri, and F. Scotti, Accuracy versus Comlixty in RBF Neural Networks, *IEEE Instrumentation and Measurement*, 4, 2001, pp. 32-36.
- [13] T. Gawne, T. Kjaer. and M. Richmond, Latency: Another potential code for feature binding in striate cortex, *J. of Neurophysiology*, 76, 1996, pp. 1356-1360.
- [14] J. O'Keefe and M.Reece, Phase relationship between hippocampal place units and the EEG theta rhythm. Hippocampus, 3, 1993, pp. 317–330.
- [15] S. Thorpe and M. Imbert., Rapid visual processing using spike synchrony. *In Advances in neural processing systems*, 9, 1996, pp. 901-907, MIT Press, Cambridge.
- [16] J. Moody, and C. Darken, Fast learning in networks of locally-tuned processing units, *Neural Comp*, 1, 1989, pp. 281-294.
- [17] D. Broomhead, and D. Lowe, Multivariable functional interpolation and adaptive networks, *Complex Systems*, 2, 1988, pp. 321-355.
- [18] S. Haykin, Neural Networks: a comprehensive foundation, Prentice Hall, 1994.
- [19] C. Bishop, Neural Networks for Pattern Recognition, Clarendon Press, Oxford, 1995.
- [20] J. Hopfield, Pattern recognition computation using action potential timing for stimulus representation, *Nature*, 376, 1995, pp. 33-36.
- [21] S. Buck, RBF++, Munich University of Technology, Software Library, Ver. 1.2, 2002.
- [22] T. Natschlager, T. and R. Ruf., Spatial and temporal pattern analysis via spiking neurons Network, *Computation in Neural Systems*, 9, 1998, pp. 319-332.
- [23] K. Negm, Real Time UNIX Kernel Based Intrusion Detection System, Proc. of International Conference of Intelligent Systems and Control. 2002, pp 126-138.
- [24] K. Negm, Double Step Intrusion Detection System, Proc. of International Conference of Intelligent Systems and Control, 2003, pp 26-32.

K. Negm, Ph.D., SMIEEE, CISSP, CISA, Associate Professor in Etisalat College of Engineering and Senior Security Specialist Dr. Negm is a member of the Information Systems Security Association (ISSA)-USA and Information Systems Audit and Control Association (ISACA)-USA. He is the Associate Chairman for the Security Standards Committee and Secretary for the Scientific Committee of the ISSA for the Middle East and Asia. Also he is a member of the Technical Committee of Security Standards of the IEEE and the USENIX group. He is a member of many IEEE committee on Security and Privacy, Task Force on Information Assurance.

Currently he is an Associate Professor in Etisalat College of Engineering, UAE. He have various International collaborations, TRIUMF-Canada, ICTP-Italy, and ECT*-Italy, NATO -Italy. For the last 18 years he have been involved in carrying out responsibilities for the Network Security Architecture, including the design, implementation, and administration of firewalls, Web servers, proxy servers, SecureID and other network security components for several Governmental Departments, Security Agencies, Banks and Educational Institutes. He provided training and consulting in the areas of security solutions and security audits. This involved writing the corporate security policy, designing and implementing the corporate firewall solution, and providing secure access for remote systems. Dr.

Dr. Negm has authored over 60 papers in refereed technical journals and international conferences. He is a Senior Member of the IEEE and Member of the Applied Computational Society. He is a regular reviewer for Modeling and Simulation Journal, IEEE Security and Privacy and Computer Security Journal.

Dr. Negm is an author of well knows published three IDS based on neural networks algorithms. Currently he is interested in IPSEC, Wireless Security, IT Forensics and the AAA Wireless Problems. Dr. Negm is listed in Who's Who in Information Technology and Networks Systems Security and Nominated to be the Professional of the Year 2004 (of IT Security) by the International Association of Networking Professionals-USA.