Tenth Int. Workshop on Generalized Nets
Sofia, 5 December 2009, 63-69

Intuitionistic Fuzzy Estimation of Damaged Packets with Multilayer Perceptron

Ivelina Vardeva and Sotir Sotirov
University of „Prof. d-r Assen Zlatarov" – Bourgas, Bulgaria,
e-mails: *iveto@btu.bg*, *ssotirov@btu.bg*

Abstract: The Ad Hoc networks are very important for provided LAN services. Ad hoc use wireless communication and very often have problem for creation and support of the transfer of the information. A purpose of the present work is the development of the neural network giving intuitionistic fuzzy estimation for reliability of the connection.

Keywords: Generalized nets, Intuitionistic fuzzy sets, Local area network, Multilayer perceptron

1 Introduction

Ad hoc networks [7] formed by randomly deployed self-organizing wireless nodes have a wide range of applications, such as tactical communications, disaster relief operations and temporary networking in sparsely populated areas, and therefore they have been studied extensively for two decades. More recently, sensor networks have attracted interest from the research community and industry. They are more energy constrained and scalable ad hoc networks.

Another form of ad hoc network, namely mesh networks, is aimed at application areas such as infrastructureless network scenarios for developing regions, low-cost multihop wireless backhaul connections and community wireless networks. Characteristics such as wireless access, mobility, rapid and random deployment make these kinds of network a very challenging field for security. Security is also a key issue in making many ad hoc application scenarios practical. Although security for these networks has been studied extensively for more than a decade, there are still many challenges waiting for better solutions. Therefore, many researchers and engineers from both academia and industry continue working on this topic.

The simplest technique for detecting an error is adding parity bits to the transmitted data. Parity bits introduce very low overhead and do not require much computing. However, they may not detect burst errors. When two or more bits are changed during transmission, they may cancel each other and the receiver may not detect the fact that the frame is garbled. Therefore, another technique called the *cyclic redundancy check (CRC)* is often used for error detection.

In this technique, first of all a bit string called the *generator polynomial* is determined. It is called the generator polynomial because CRC is treated as a polynomial operation where input and generator bit strings are represented as polynomials with coefficients of 0s and 1s. A generator polynomial is used to generate *checksums*, which are appended at the end of frames. The receiver checks the input by using the same generator to detect transmission errors. There are two rules in selecting generator polynomials:

- They must be shorter than the frame length;
- They must start and end with 1.

If the text is attacked on its way to the receiver and some bits are changed, the corresponding bits in the decrypted plain text will also be changed. Also, if an attacker can capture two cipher texts encrypted using the same key stream and XOR these two cipher texts,then this result is also the XOR of the two appropriate plain texts. Having gained and collected this information, the attacker can use statistical attacks to recover the plain texts. The more frequently a key stream is used for encryption and is captured by an attacker, the more easily he can perform statistical attacks. When the attacker recovers one of the plain texts, he can also recover the others.

The integrity check field referred to above is a checksum with 32-bit CRC and is also encrypted.

Intuitionistic Fuzzy Set

The sets showing the degree of probability μ and the degree of improbability ν whether one connection is reliable or not, are represented by the ordered pair $<\mu, \nu>$ of real numbers from the set $[0,1]\times[0,1]$, see [3]. The degree of uncertainly $\pi = 1-\mu-\nu$ presents the cases when the information from/to other computer improbability is insufficient. In such cases, there is a need of additional information. The ordered pairs are determined in the meaning of the theory of intuitionistic fuzzy sets, [3].

In that case is good to use a neural network giving at its exits intuitionistic fuzzy sets μ, ν and π.

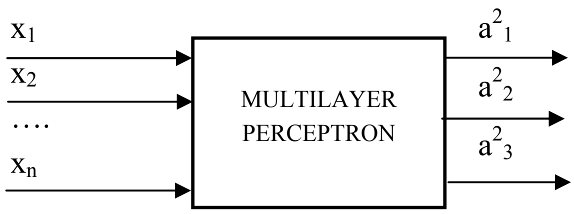

Figure 1

The first of the exits gives a set about the stage of probability the connections to be good - μ. The second of the exits gives a set about the stage of improbability connections to be good - ν, and the third one gives a set about the stage of uncertainly $\pi = 1-\mu-\nu$.

In submitting at the entries of the network (Figure 1) the standard composed letter (with normal fields, data and CRC) at the exit of the neural network is received minimum stage of probability the connections to be good - μ and maximum stage of improbability the connections to be good - ν.

Then, when is received a damaged e-mail is get a maximum value of the stage of uncertainly π but for account of μ and ν.

Neural network

In the reference sources different types of neural networks are described. Some of them are used for identification of classes (symbols, types of damages, knowledge and so on).

In the realization a classic neural network with right transmission is used (Figure 2). It is learnt in an accelerated variant of the backpropagation algorithm.

At its entry values are submitted for the sixth separate parameters described in the upper rows. The network is learnt so that to yield intuitionistic fuzzy sets μ, ν and π at its exit.

The neural network is learnt for 33 epochs in the environment of MATLAB. Assigned average square error is 1.10^{-4} but is achieved – $6.8637.10^{-5}$ (Figure 3).

Learning

The learning of the neural network is an essential part of the whole system. The entry vectors and purposes by which the net is learnt, are selected so that they have to show in which entry influences what is received at the exit of the network.

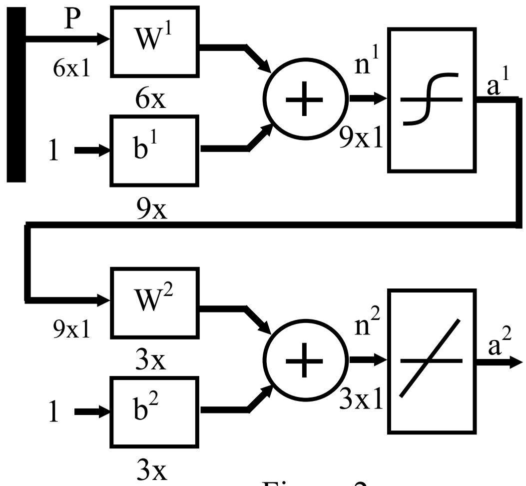

Figure 2

In this case we use a slightly different way than the classical one. At the beginning a statistics is drawn of the different types of connections and after that the neural network is learnt.

At the beginning when the information has not been extracted yet, every set receives the value <0, 0>. The current $(k+1)^{-st}$ set for $k \geq 0$ is estimated on the basis of the previous sets from the formula:

$$<\mu_{k+1}, \nu_{k+1}> = < \frac{\mu_k k + m}{k+1}, \frac{\nu_k k + n}{k+1} >,$$

where $<\mu_k, \nu_k>$ is the previous set, $<m, n>$ is the set of the last message. For $m, n \in [0, 1]$ and $m + n \leq 1$. So, at the mail server it is formed a set for the probability and improbability for one connection to be a good one.

Testing

In the process of testing, 10 test vectors are submitted at the entry of the neural network combined, so that they exhibit the different characteristics of the system.

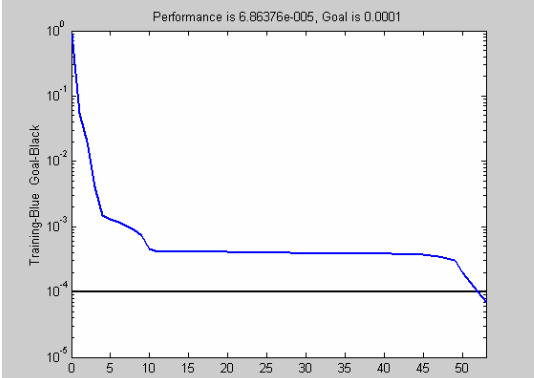

Figure 3

The first test vector responds to the normal status of the neural network. In it, each of the separate components is in normal range.

The next four vectors present the different types of inputs of the neural network. These different types of inputs exhibit different errors for different field types of the WEP packet format.

Next three vectors are from that type, in which there is passing each other in two of the common components. In the third test vectors they are at the different places.

The last two test vectors are selected so that to have more than 2 passings.

Table 1

№	Inputs on the neural network					Outputs		
						μ	ν	π
1	0	0	0	0	0	0.0006	0.9979	0.0005
2	1	0	0	0	0	0.1468	0.8040	0.0459
3	0	1	0	0	0	0.1501	0.7991	0.0499
4	0	0	1	0	0	0.1499	0.8004	0.0499
5	0	0	0	1	0	0.1503	0.7989	0.0504
6	0	0	0	0	1	0.6610	0.1849	0.1030
7	1	0	0	0	1	0.7268	0.1748	-0.0334
8	0	0	1	0	1	0.7559	0.1607	0.1131
9	1	1	0	1	0	0.8657	0.1658	0.0927
10	1	0	0	1	1	0.8425	0.1512	0.0296

Results

Based on the researches, the following deductions can be derived:

- The neural network widens the standard system for detecting of good communication using the Ad Hoc network;
- The received system gives a probability set about whether the respective communication is good or not;
- In submitting of zero values at the entry of the network (row 1 from table 1) the values for the degree of probability the connections is good and uncertainly are little, and the value for the improbability – bigger;
- In reading the separate random values of one of the properties only, it is a small probability that the connections are good (rows 2-5 from the table1);
- In submitting of values showing two changes of the signs for good communication (rows 6-8 from the table1) at the exit of the network the values of μ increase and those of ν and π decrease;

- In submitting of values showing three (and more) signs good communication (rows 9-10 from the table 1) at the exit of the network for the values of the probability it could communication be a good are ≈1 and for improbability and uncertainly decrease almost to zero.

GN-model

All definitions related to the concept "GN" are used from [1, 2].

Initially the following tokens enter the GN net:

- one α_F-token with characteristic "Data for learning couples" from place L_1;
- one α_n-token with characteristic "Structures of the MPLs" from place L_5;
- one α_e-token with characteristic "Square error" from place L_{10}.

The generalized net is introduced by the set of transitions:

$$A = \{ Z_1, Z_2, Z_3, Z_4, Z_5\},$$

where the transitions describe the following processes:

Z_1 – Preparing the learning couples;
Z_2 – Designing the multilayer percepron (MPL);
Z_3 – Calculating the exit MPL;
Z_4 – Process of the determination for trained MPL ;
Z_5 – Testing the test vectors.

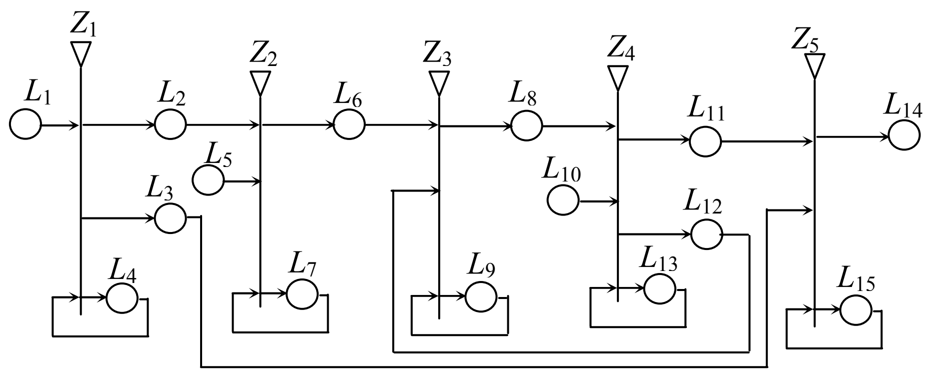

Fig. 4. GM model for process for producing Intuitionistic fuzzy estimation of damaged packets with multilayer perceptron

The transitions of GN–model have the following description.

$$Z_1 = \langle\{L_1, L_4\}, \{L_2, L_3, L_4\}, R_1, \wedge(L_1, L_4)\rangle,$$

$$R_1 = \begin{array}{c|ccc} & L_2 & L_3 & L_4 \\ \hline L_1 & false & false & true \\ L_4 & W_{4,2} & W_{4,3} & false \end{array},$$

where:

$W_{4,2}$ = "The learning pairs are prepared";

$W_{4,3} = W_{4,2}$.

The α_F-token that enters place L_4 do not obtains new characteristic. It splits into two new tokens - α_F' and α_F" that enter places L_2 and L_3 with obtain characteristic

"Learning pairs".

$$Z_2 = \langle\{ L_2, L_5, L_7 \}, \{L_6, L_7 \}, R_2, \vee (L_2, L_5, L_7) \rangle,$$

$$R_2 = \begin{array}{c|cc} & L_6 & L_7 \\ \hline L_2 & false & true \\ L_5 & false & true \\ L_7 & W_{7,6} & true \end{array},$$

where:
$W_{7,6}$ =”the MPL is designed”.

The α_F’ and α_n-token that enter place L_7 merge in a α_{n1}-token. On the next activation of the transition Z_2 the α_{n1}-token splits into two tokens - α_{n1}’- and α_{n1}”-token. The original token continue its staying in place L_7 while the new α_{n1}’-token enters place L_7 with the characteristic

“Designed MPL”.

$$Z_3 = <\{ L_6, L_{12}, L_9\}, \{L_8, L_9\}, R_3, \wedge (L_6, L_{12}, L_9)>,$$

$$R_3 = \begin{array}{c|cc} & L_8 & L_9 \\ \hline L_6 & false & true \\ L_{12} & false & true \\ L_9 & W_{9,8} & true \end{array},$$

where
$W_{9,8}$= “ exit of the MPL”.

The α-tokens that enter place L_9 merge in a α_A-token. On the next activation of the transition Z_3 the α_A-token splits into two tokens - α_A’- and α_A”-token. The original token continues its staying in place L_9 while the new α_A’-token enters place L_8 with the characteristic

“Exit of the MPL”.

$$Z_4 = <\{ L_8, L_{10}, L_{13}\}, \{L_{11}, L_{12}, L_{13}\}, R_4, \wedge (L_8, L_{10}, L_{13})>,$$

$$R_4 = \begin{array}{c|ccc} & L_{11} & L_{12} & L_{13} \\ \hline L_8 & false & false & true \\ L_{10} & false & false & true \\ L_{13} & W_{13,11} & W_{13,12} & true \end{array}$$

where
$W_{13,11}$= “square error $\leq$ calculated error ”;
$W_{13,12}$=“square error > calculated error ”.

The α-tokens that enter place L_{13} merge in a α_{A1}-token. On the next activation of the transition Z_4 the α_{A1}-token splits into three tokens - α_{A1}’-, α_{A1}”-and α_{A1}”’-token. The original token continues its staying in place L_{13} while the other α_{A1}-tokens enter places L_{11} and L_{12} with the characteristic respectively:

“Learned MPL”

in place L_{11},

and “Non learned MPL”

in place L_{12}.

$$Z_5 = <\{L_{11}, L_3, L_{15}\}, \{L_{14}, L_{15}\}, R_5, \wedge (L_{11}, L_3, L_{15})>,$$

$$R_5 = \begin{array}{c|cc} & L_{14} & L_{15} \\ \hline L_{11} & false & true \\ L_3 & false & true \\ L_{15} & W_{15,14} & true \end{array}$$

where

$W_{15,14}$= "test results are calculating ".

The α-tokens that enter place L_{15} merge in a α_{A2}-token. On the next activation of the transition Z_5 the α_{A2}-token splits into two tokens - α_{A2}'- and α_{A2}"-token. The original token continues its staying in place L_{15} while the other α_{A2}-token enters place L_{14} with characteristic

"Result for the test vectors".

Conclusion

The Ad Hoc networks are very important for provided LAN services. Ad hoc use wireless communication and very often have problem for creation and support of the transfer of the information. A purpose of the present work is the development of the neural network giving intuitionistic fuzzy estimation for reliability of the connection. In the end of the paper we develop GN model. It can used for investigating the process in the future.

References

1. Atanassov K., On Generalized Nets Theory, "Prof. M. Drinov"Academic Publishing House, Sofia, 2007
2. Atanassov, K. Generalized Nets, World Scientific. Singapore, New Jersey, London, 1991
3. Atanassov, K. Intuitionistic Fuzzy Sets, Springer, Heidelberg, 1999
4. Haykin, S. "Neural Networks: A Comprehensive Foundation", Prentice Hall, N.J., 1999
5. Bishop C. M., Neural networks for pattern recognition, Oxford university press, ISBN 0 19 853864 2 , 2000
6. M.T.Hagan, H.B.Demuth, M.Beale, "Neural Network Design", PWS Publishing Company, Boston, 1996
7. Erdal Çayırcı, Chunming Rong, Security inWireless Ad Hoc and Sensor Networks, 2009 John Wiley & Sons Ltd.