Recognition of Airplane Wing-Tip Vortices Encounters Using Neural Networks

Aziz Al-Mahadin and Faouzi Bouslama

***Abstract*—In earlier papers, the authors identified critical parameters to be used in any effective identification of aircraft vortex encounters. Various techniques of pure fuzzy logic and hybrid soft-computing approaches were used to model and successfully classify vortex encounters. In this paper, the authors consider pure neural networks models having different architectures to identify aircraft encounters of wing-tip vortices. The automatic identification of airplane vortex encounters using neural networks gives excellent accuracy when compared with manual approaches. The highest accuracies are obtained by probabilistic neural networks. They are about 93%, 73% and 83% for the overall training, the overall testing and the overall average, respectively. The achieved results confirm the effectiveness of some neural network techniques and the choice of the critical parameters to automatically identify wing-tip vortices.**

***Index Terms*—Wing tip vortices, vortex encounters, neural networks (NN), flight data records.**

I. Introduction

Recent studies on the automatic identification of aircraft vortex encounters have shown a great potential in using soft-computing approaches. In [1], the authors used fuzzy logic (FL) to model and identify vortex encounters. FL tolerates data imprecision and cope well with complexities in modeling the vortex encounters. Fuzzy linguistic variables were used to model data from flight data recorders (FDRs) and pilot reports. The fuzzy rules were derived from a collection of 54 pilot reports [2] of vortex encounters and 210 records of flight events from FDRs. An average success rate of identification of 83.7% was obtained.

In [3], a neuro-fuzzy identification system was used to classify vortex encounters. Artificial neural networks integrated with fuzzy systems have been used as a solution in the automatic tuning of the membership functions of fuzzy linguistic variables and applied to various problems. The authors used a hybrid Adaptive Neuro-Fuzzy Inference System (ANFIS) to automatically tune the parameters of the fuzzy membership functions. They investigated various neuro-fuzzy models having different sets of parameters and factors, and they achieved an average identification accuracy of 84.2%.

This paper builds on the previous results achieved with FL and ANFIS approaches, respectively, in the automatic identification of aircraft vortex encounters. The authors continue to investigate machine learning by using various architectures of pure neural networks (NNs). These NNs are constructed based on a similar reduced set of parameters as in [1]-[3]. The paper is structured as follows. Section II introduces the airplane wing tip vortex problem including the selection of the critical parameters relevant to this investigation. Section III explains the identification classes and input vectors. Section IV provides details on the proposed NNs investigated in this research. Section V is the conclusion.

Manuscript received October 18, 2018; revised December 26, 2018.

The authors are with the Dubai Men’s College, Higher Colleges of Technology, Dubai, UAE (e-mail: aziz.almahadin@hct.ac.ae, faouzi.bouslama@hct.ac.ae).

II. Wing Tip Vortices and Critical Parameters

All airplanes generate wing tip vortices due to the pressure difference between wing upper and lower surfaces, Fig. 1. This vortex may cause danger to following aircraft [4]. Hence, it is important to identify actual vortex encounters in order to introduce mitigation measures.

The potential hazard of a vortex on a following airplane can vary depending on a number of parameters such as the type of following and the leading airplanes, the flight phase, the airplane weight, the wing size, airplane configuration and the weather conditions. Encountering vortex can be hazardous during flight, in particular, at landing and takeoff flight phases, where the airplanes are required to fly within constrained flight paths, which makes vortex encounter avoidance and recovery more difficult [2] and [4].

Fig. 1. Airplane wing tip vortices.

For the purpose of this research, 181 FDR records are collected which were reported to contain vortex encounters. In addition, another 29 records are utilized which were reported to contain other flight events such as wind shear and hard landing [2]. These later records are used to compare flight events and to test the appropriateness of the various NN techniques to discriminate vortex encounters from other flight events.

FDRs contain over one thousand parameters, but only 8 are found to be relevant to this investigation, Table I [2].

doi: 10.18178/ijmlc.2019.9.2.774

TABLE I: INPUT PARAMETERS RELEVANT TO VORTEX ENCOUNTERS

INPUTS
Normal Acceleration
Lateral Acceleration
Derived Normal Acc. Rate
Roll Angle
Derived Roll Rate
Control Wheel
Derived G-Time
Derived Roll Time

III. SELECTION OF CLASSES AND INPUT VECTORS

In this research, two types of models are considered: a 2-class model which contains either points to vortex encounter or not, and a 3-class model which additionally includes a third class indicating a possible vortex encounter. Class 1 always indicates vortex encounter, Class 2 indicates a possible vortex encounter, and Class 3 indicates a non-vortex encounter with probability values of 1, 0.5 and 0, respectively [2], as shown in Table II. Number of input vectors are larger than records since some FDR records contain more than one vortex or flight event.

TABLE II: INPUT VECTORS DISTRIBUTION ON THE VARIOUS CLASSES

	Number of vortex records	Input vectors			
		3-class model		2-class model	
		Training	Testing	Training	Testing
Class 1	100	125	10	175	12
Class 2	39	70	8	-	-
Class 3	42	113	23	133	29
Total	181	308	41	308	41

The evaluation of the classification capability of the various networks is based on three values which are calculated as follows:

$$\%\mathrm{Class}N = \frac{\text{Number of records correctly classified in class } N}{\text{Total records in class } N} \times 100\% \quad (1)$$

$N = 1, 2, 3$

$$\%\mathrm{Overall} = \frac{\text{Number of records correctly classified from all classes}}{\text{Total number of records}} \times 100\% \quad (2)$$

$$\%\mathrm{Overall\ Avg} = \frac{\%\text{Overall taining data} + \%\text{Overall testing data}}{2} \times 100\% \quad (3)$$

The Matlab NN toolbox [5] is used to simulate the various architectures. NN modeling is made more efficient by using normalization techniques [5] such as scaling the inputs to fall within the [-1, 1] range. Normalized vectors derived by using this technique are referred to as *Vn*. Another approach for scaling network inputs is to normalize the inputs and targets to have zero mean and unity standard deviation. Normalized vectors derived by using this technique are referred to as *Vs*. In some situations, the dimension of the input vectors is large, but the components of the vectors are highly correlated and, hence, can be reduced using principal components analysis [5]. Two sets of parameters are used with this technique to eliminate those principal components that contributed less than 2% (*Vp*2) and 0.1% (*Vp*1) to the total variation in the data sets. Un-normalized input vectors are referred to as *V*.

IV. NEURAL NETWORKS USED IN THE IDENTIFICATION OF VORTEX ENCOUNTERS

There are many types of neural networks that differ in their features, complexity, learning algorithm, structures and applications. The proper selection of a neural network depends largely on the application [6]. Six types of NNs are investigated: perceptron, linear, radial basis, probabilistic, linear vector quantization (LVQ), and Elman.

A. *Perceptron Neural Networks*

A perceptron network has a single layer of hard-limit transfer functions. Such networks are used as classifiers for linearly separable input vectors [7] and [8]. The perceptron neural network is investigated to classify the vortex data into two classes: vortex encounter and non-vortex encounter. Fig. 2 and 3 show the results of investigating various parameters.

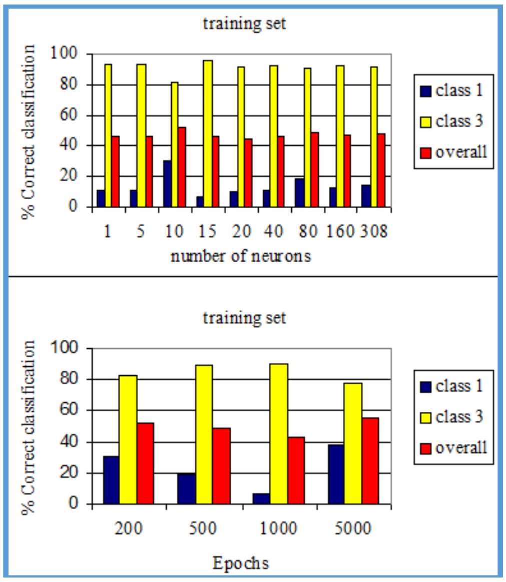

Fig. 2. Perceptron network investigation results.

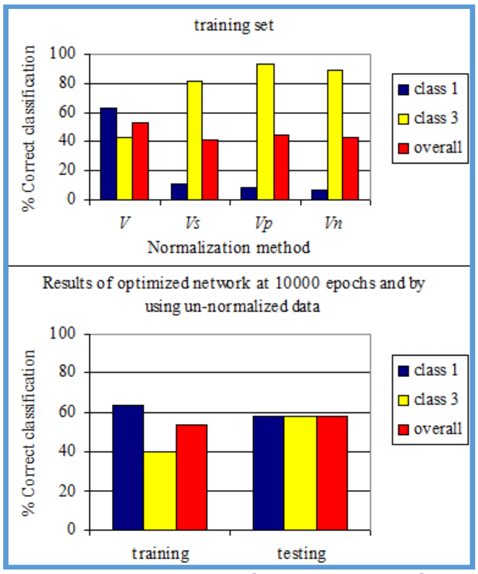

Fig. 3. Perceptron network investigation results.

The upper plot in Fig. 2 shows that the network gives the best result at 10 neurons with a classification accuracy of 32%, 82% and 52% for class 1, class 3 and overall accuracy respectively. The lower plot in Fig. 2 shows that different accuracies are obtained at various epochs with the highest obtained at 5000. The average simulation time is found to be about 1 second per epoch (each traverse through all of the training input and target vectors is called an epoch or pass). Investigation of normalization techniques revealed that un-normalized data gave the best results, as shown in the upper plot in Fig. 3.

The optimized parameters (i.e. 10 neurons and the un-normalized data) are used at 10,000 epochs which gives accuracies of about 63%, 40% and 53% for class 1, class 3 and overall, respectively, for the training data. Testing data is classified with accuracies of 58%, 59% and 59% for class 1, class 3 and overall, respectively.

B. Linear Neural Networks

Linear networks [9], [10] are used to identify vortex encounters by adjusting number of parameters including the number of neurons, the learning rate and by selecting the appropriate normalization technique [5]. It is found that the number of neurons has no effect on the classification accuracy with disregard to all other parameters as shown in Fig. 4.

As for the learning rate (*α*), it is found that any α equal or greater than 0.00001 results in high error with disregard to all other parameters for both the 2 and 3-class models. Other parameters are also investigated, such as the normalization technique where it is found that un-normalized data produced the highest classification percentage. Furthermore, it is found that the 2-class model produced higher correct classifications of about 69%, 51% and 61% for class 1, class 3, and overall, respectively. However, the linear network does not seem to be appropriate for this problem with the current data and inputs due to its low classification accuracy.

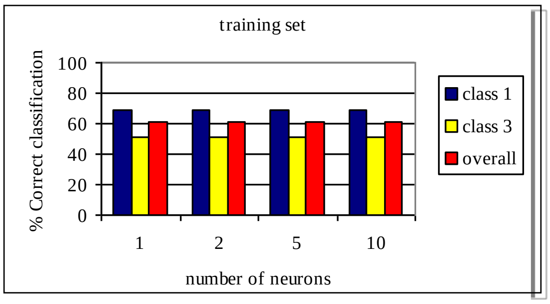

Fig. 4. Effect of number of neurons on classification accuracy.

C. Radial Basis Neural Networks

Radial basis networks consist of two layers: a hidden radial basis layer and an output linear layer [11] and [12]. Various parameters are investigated including normalization method. Fig. 5 and 6 show the effect of normalization method on classification accuracy of the 2 and 3-class models respectively.

The training set has better classification accuracy especially for the 2-class model which is about 100% for both class 1 and 3 while using the normalization vectors *V*, *Vs* and *Vp1* as shown in Fig. 5. The 3-class model also has high classification accuracy for the training set, Fig. 6. However, both models show that the testing data is poorly classified. This problem is known as lack of generalization [13].

The second parameter which is investigated for this network is the spread constant which determines the width of an area in the input space to which each neuron responds [5]. For example, if spread constant is 6 then each neuron will respond to any input vectors within a vector distance of 6. Therefore, it should be large enough so that neurons respond to overlapping areas. It is found that the lowest spread constant to satisfy all classes and both training & testing sets is 24 as in Fig. 7.

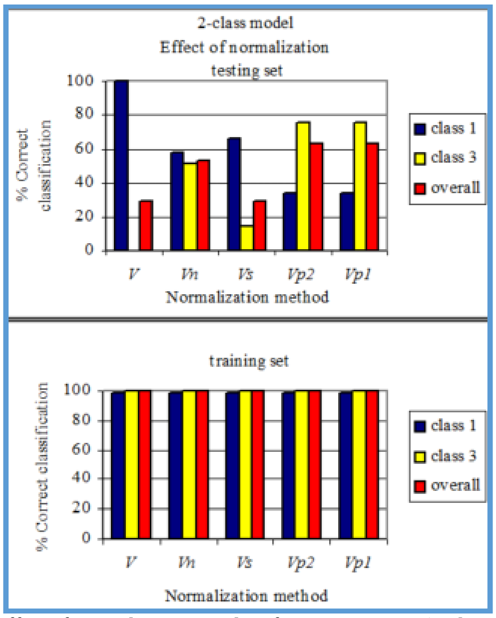

Fig. 5. Effect of normalization on classification accuracy (2-class Model).

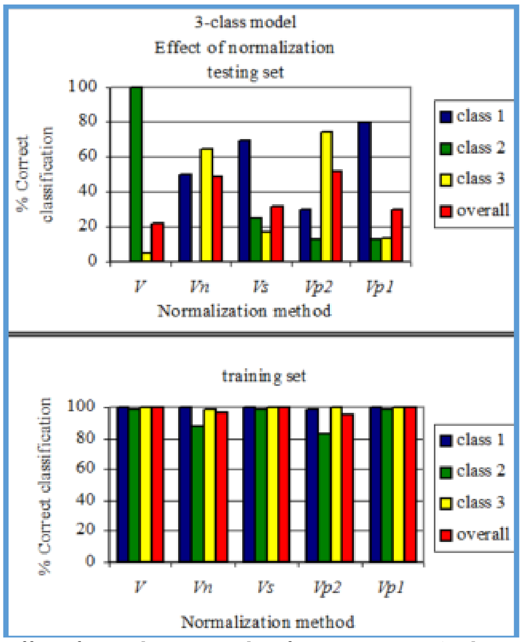

Fig. 6. Effect of normalization on classification accuracy (3-class Model).

The NN toolbox [5] offers two radial basis design functions (*newrbe* and *newrb*) where the former creates as many neurons as there are input vectors while the latter finds the smallest network that can solve the problem within a given error goal by creating one neuron at a time and continues to add neurons until error falls under an error goal or a maximum number of neurons has been reached. Fig. 8 shows a comparison of these two functions where *newrbe* gives higher percentages for class 2 and class 3 of the testing set.

Other parameters such as the goal (desired minimum error) also is considered and showed no effect on the identification accuracy. The optimized parameters (i.e. un-normalized data *V*, spread constant of 24 and the design function *newrbe*) are used in one radial basis neural network and the results are shown in Fig. 9. It is clear from the figure that both models failed to identify class 2. Therefore, the combination of optimized parameters does not necessarily produce an optimized network.

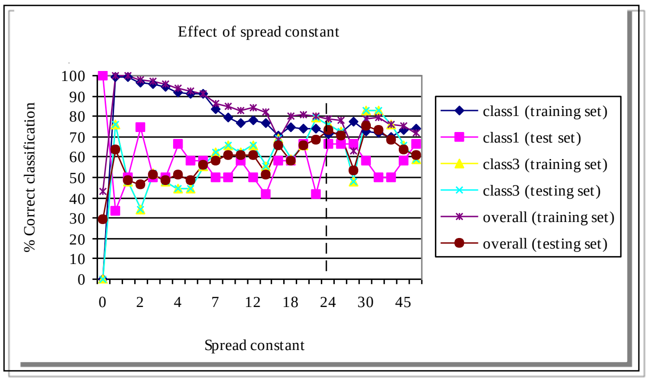

Fig. 7. Effect of spread constant on classification accuracy.

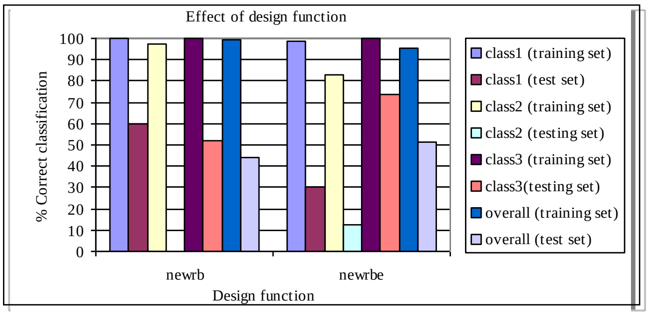

Fig. 8. Effect of design function on identification accuracy.

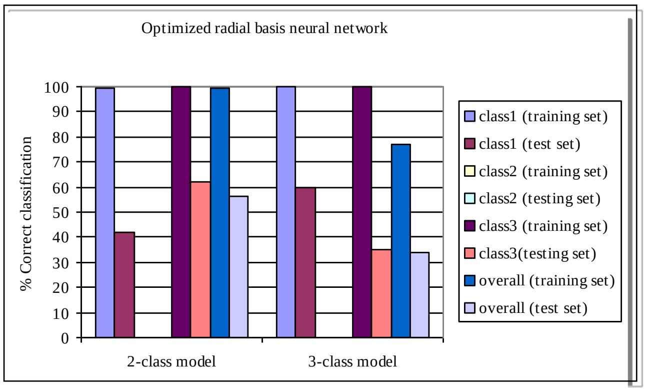

Fig. 9. Identification accuracy using the optimized radial basis parameters.

D. Probabilistic Neural Networks

Probabilistic neural network architecture is similar to the radial basis network except that the second layer is a competitive one instead of the linear layer. When an input is presented, the first layer computes distances from the input vector to the training input vectors, and produces a vector whose elements indicate how close the input is to the training inputs. The second layer sums these contributions for each class of inputs to produce, as its net output, a vector of probabilities. Then the transfer function of the second layer picks the maximum of these probabilities, and produces a 1 for that class and a 0 for the other classes [5], [12] and [14].

This type of networks produced better classification and generalization than the radial basis network, as shown in Fig 10 and 11. It is clear from the figure that the 2-class model produced more accurate identification for both classes using the data set *Vn* where the highest obtained accuracies are about 93%, 73% and 83% for overall training, overall testing and overall average respectively.

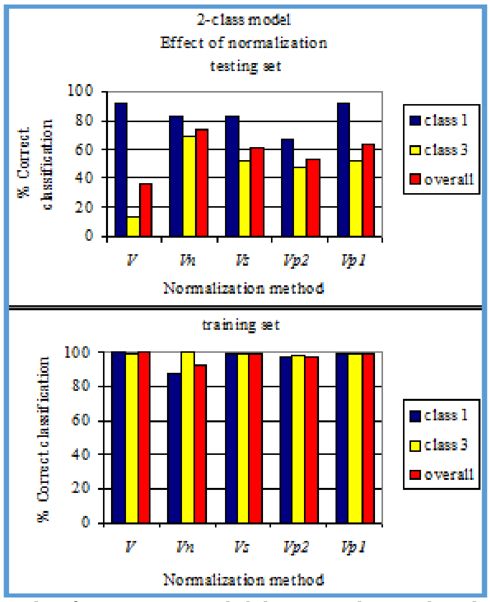

Fig. 10. Results of investigating probabilistic neural network/2-class Model.

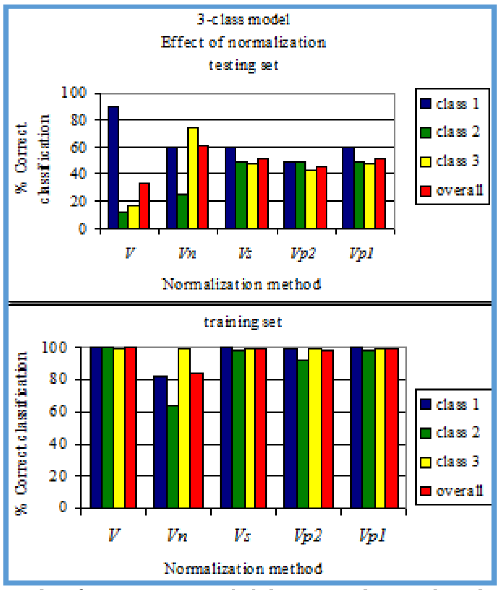

Fig. 11. Results of investigating probabilistic neural network/3-class Model.

E. Learning Vector Quantization Networks

Learning vector quantization (LVQ) networks classify input vectors into target classes by using a competitive layer to find subclasses of input vectors, and then combining them into the target classes. An LVQ network has a competitive layer and a second linear layer. The competitive layer learns to classify input vectors into subclasses then the linear layer transforms the competitive layer's classes into target classes defined by the user [5] and [15].

The LVQ network is considered using various values of network parameters as shown in Fig. 12 and 13. Fig. 12 shows that *Vn* has only a slight increase in classification accuracy with respect to all other techniques. Increasing the number of neurons above 2 does not significantly improve accuracy. This is clear in Fig. 12 where the overall accuracy at 2 neurons and above is around 60%. However, both figures show that the network performed poorly, since it put most of the data in one class and only a small percentage (less than 14%) in the other class.

The learning rate (*lr*) is investigated and revealed no improvement in network classification. As shown in Fig. 14, the network gave only a small percentage of correct classification for class 3. Various other network settings are tested for the 2-class model. The best combination is obtained using *Vn*, 12 neurons, 0.01 learning rate and 50000 epochs which gave accuracy of 94%, 20% and 62% for class1, class3 and overall respectively. Similar investigation is carried out for the 3-class model where the highest classification is found to be 71%, 1.4%, 56%, and 50% for class 1, class 2, class 3 and overall respectively. Therefore, the LVQ network is not satisfactory, but, it could possibly be improved with further investigation and increase of the training and testing data.

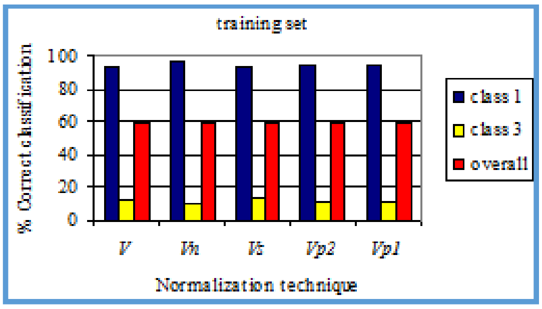

Fig. 12. Results using LVQ and various normalization techniques.

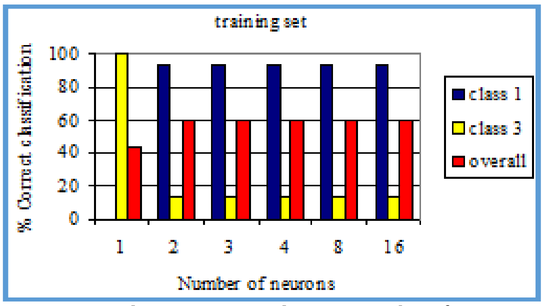

Fig. 13. Results using LVQ and various number of neurons.

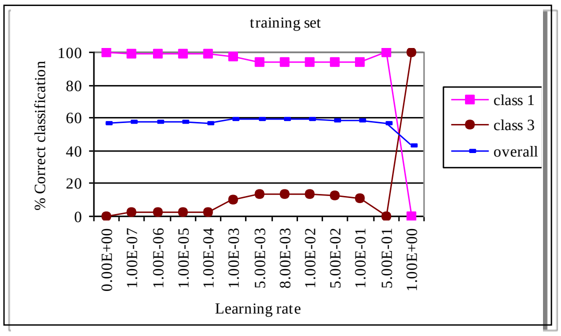

Fig. 14. Effect of learning rate on classification of LVQ network.

F. Elman Networks

The Elman network has two layers with a feedback connection from the output of the first layer to its summation junction as shown in Fig. 15. The delay in this connection stores values from the previous time step, which can be used in the current time step. Therefore, even if two Elman networks, with the same weights and biases, are given identical inputs at a given time step, their outputs can be different due to different feedback states [16] and [17]. Thus, the Elman network is able to learn temporal patterns as well as spatial patterns.

One of the parameters investigated using the Elman network is the number of neurons in the first layer. Fig. 16 and 17 show that the network failed to classify the training and the testing data and put most of the data in class 1. The maximum correct classification of class 3 in the training data is 20% with 150 neurons. Furthermore, it is found that the training time is very high, even at low epochs as shown in Fig. 18. When the number of epochs is increased to 6425, the time increased to 10 hours.

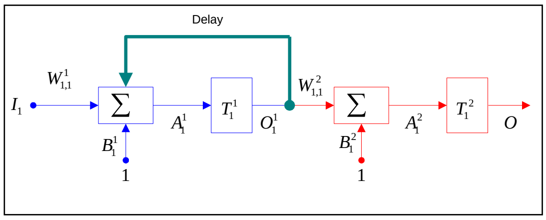

Fig. 15. A two-layer Elman network.

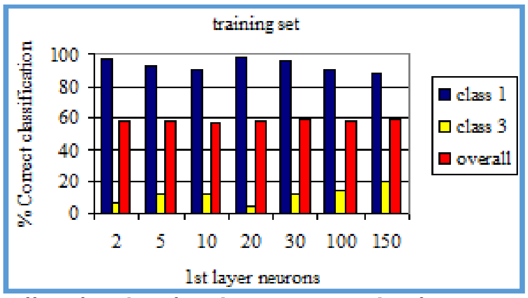

Fig. 16. Effect of number of 1st- layer neurons on classification/training set.

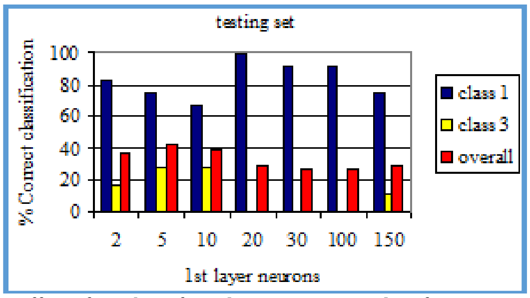

Fig. 17. Effect of number of 1st- layer neurons on classification/testing set.

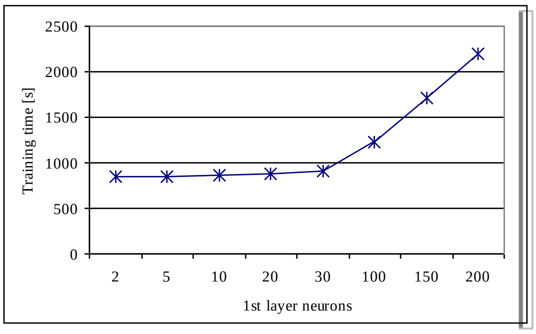

Fig. 18. Training time at 100 epochs versus number of first layer neurons.

Other network parameters, such as type of transfer functions and normalization techniques are also investigated and it is found that the maximum classification accuracy for class 3 is 20%, therefore concluding the unsuitability of this network.

V. Conclusion

This paper presented the results of investigation of pure neural networks in the automatic identification of aircraft vortex encounters. Six various types of neural networks are considered. Only the critical roll angle was used as parameter of identification. The probabilistic NNs are found to be the most appropriate achieving about 93%, 73% and 83% for the overall training, the overall testing and the overall average, respectively. This finding is explained based on the fact that probabilistic NN is more appropriate for classification and pattern recognition problems compared to other NN's. Furthermore, it is insensitive to outliers due to its structure and algorithm. ANFIS techniques give slightly better identification compared to FL and all other investigated NN's. This can be justified since ANFIS allows human interventions to select the appropriate fuzzy rules which is done based on familiarity with the vortex problem. In addition, it deals well with data imprecision and complexities in modeling vortex encounters.

In future work, the normal and lateral accelerations will be considered as inputs and more vortex encounter records will be utilized to improve identification.

Acknowledgment

The authors convey their gratitude to the support received from Dubai Men's College, the Higher Colleges of Technology, UAE, in conducting this research.

References

[1] A. Al-Mahadin and F. Bouslama, "Automatic identification of wake vortex traverse by transport aircraft using fuzzy logic," in *Proc. the IEEE 4th Intl. Conference on Soft Computing & Machine Intelligence*, Mauritius, Port Louis, 2017, pp. 133-139.

[2] A. Al-Mahadin and S. Dalkilic, "Investigation of factors affecting aircraft vortex encounters," in *Proc. the IEEE 2018 Advances in Science and Engineering Technology International Conferences (ASET)*, Dubai, UAE, 2018, pp. 1-6.

[3] A. Al-Mahadin and F. Bouslama, "Neuro-fuzzy techniques for the identification of aircraft wake vortex encounters," in *Proc. IEEE 2018 Advances in Science and Engineering Technology International Conferences (ASET*), Dubai, 2018, pp. 1-6.

[4] F. Holzapfel and M. Steen, "Aircraft wake-vortex evolution in ground proximity: analysis and parametrization," *AIAA Journal*, vol. 45, no. 1, January 2007.

[5] *Neural Network Toolbox*, Math Works Documentation, Math Works Inc., 2017.

[6] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Prentic-Hall, Inc., New Jersey, 1999.

[7] M. Hagan, H. Demuth, M. Deale and O. Jesus, *Neural Network Design*, 2nd ed. MIT Press, MA, 1989.

[8] M. Gardnera and R. Dorlinga, "Artificial neural networks (the multilayer perceptron) — A review of applications in the atmospheric sciences," *Atmospheric Environment*, vol. 32, issues 14–15, pp. 2627-2636, August 1, 1998.

[9] E. Oja, "Principal components, minor components, and linear neural networks," *Neural Networks*, vol. 5, issue 6, pp. 927-935, Nov.–Dec. 1992.

[10] P. Baldi and K. Hornik, "Lerning in linera neural networks: A survey," *IEEE Transactions on Neral Networks*, vol. 6, issue 4, pp. 837-858, July 1995.

[11] M. Han and J. Xi, "Radial basis perceptron network and its applications for pattern recognition," in *Proc. the 2002 International Joint Conference on Neural Networks*, pp. 669-674, vol. 1, 2002.

[12] H. De-Shuang and Z. Wenbo, "A novel method for improving the classification capability of radial basis probabilistic neural network classifiers," in *Proc. the 2002 International Joint Conf. on Neural Networks*, 2002, vol. 1, pp. 102-106.

[13] S. Urolagin and N. Reddy, "Generalization capability of artificial neural network incorporated with pruning method," *Advanced Computing, Networking and Security*, Berlin, vol. 7135, 2012.

[14] K. Mao, K. Tan, and W. Ser, "Probabilistic neural-network structure determination for pattern classification," *IEEE Transactions on Neural Networks*, vol. 11, issue 4, pp. 1009-1016, July 2000.

[15] T. Kohonen, "Learning vector quantization, in: self-organizing maps," *Springer Series in Information Sciences*, vol. 30, Berlin, 1995.

[16] E. Haselsteiner, "What Elman networks cannot do," in *Proc. IEEE World Congress on Computational Intelligence—International Joint Conference on Neural Networks*, 1998, vol. 2, pp. 1245-1249.

[17] L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, pp. 179-211, 1990.

Aziz Al-Mahadin is an assistant professor and chair of aviation engineering at Dubai Men's College, Higher Colleges of Technology, UAE. He received his PhD in aerospace engineering and artificial intelligence from Hertfordshire University, UK in 2003.

He has an MSc in mechanical engineering, an MSc in art and science of warfare, a BSc in aerospace engineering and a BSc in military science. His research interest is in the areas of aerospace and mechanical engineering, artificial intelligence with aviation applications, energy & environment, UAVs and military applications.

Dr. Aziz has held many positions in academia, industry and military.

Faouzi Bouslama is an associate professor at the Computer Information Science (CIS) Department, Dubai Men's College (DMC), the Higher Colleges of Technology (HCT), the UAE. He is also the division chair of the CIS Department at DMC. He received his Ph.D. in electronic engineering from the Graduate School of Electronic, Science and Technology, Shizuoka University, Japan, in 1992. His research interests include intelligent systems and soft computing (fuzzy logic and neuro-fuzzy systems), business intelligence and data mining, data modeling and enterprise architectures, smart learning environments, emotional intelligence and student success. Dr. Faouzi is a member of the IEEE UAE A.I. technical board and a member of TC on soft computing, IEEE systems, man and cybernetics (SMC) society.