The ANNIGMA-Wrapper Approach to Fast Feature Selection for Neural Nets

Chun-Nan Hsu, Hung-Ju Huang, and Dietrich Schuschel

Abstract— **This paper presents a novel feature selection approach for backprop neural networks. Previously, an feature selection technique known as *the wrapper model* was shown effective for decision trees induction. However, it is prohibitively expensive when applied to real-world neural net training characterized by large volumes of data and many feature choices. Our approach incorporates a weight analysis based heuristic called ANNIGMA to direct the search in the wrapper model and allows effective feature selection feasible for neural net applications. Experimental results on standard data sets show that this approach can efficiently reduce the number of features while maintaining or even improving the accuracy. We also report two sucessful applications of our approach in the helicopter maintenance applications.**

Keywords— **Feature Selection, Curse of Dimensionality, Neural Networks, the Wrapper Model**

I. Introduction

Selection of relevant features is of primary importance to the success of a neural net. The goal is to find the minimum subset of features that yield the highest accuracy. This problem is especially severe when real-world applications are attempted and human selection of features is not available, desirable, or dependable. There are two models of feature subset selection. In the filter model, the features are filtered independently of the induction algorithm. This filtering is done as a preprocessing step. In contrast, the wrapper model (John, Kohavi, Pfleger, 1994) wraps around the induction algorithm, searching the feature subset space guided by the performance of the induction algorithm.

Since the filtering model ignores the effect of the feature subset on the performance of the induction algorithm, many researchers have pointed out that it may not be as effective and general as the wrapper model [1], [2], [3]. They make the point that feature subset selection must take into account the biases of the induction algorithm in order to perform well. However, since in the wrapper model, a large number of training is required to search for the best performing feature subset, it can be prohibitively expensive for neural net applications. Many search strategies were proposed to speed up the search, including hill climbing [3], compound operators [2], randomized algorithms [4], etc. However, when applied to neural net application, at each branching point of the search, these approaches still need to train m neural nets with cross validation to select the next feature subset, where m is the branching factor. This can be prohibitively expensive in real-world applications. A directed feature subset search is needed for neural nets. In this paper's approach, the feature subset search is accelerated by a heuristic called Artificial Neural Net Input Gain Measurement Approximation (ANNIGMA). ANNIGMA ranks neural net features by relevance. Following each training, the ANNIGMA heuristic ranks the features by relevance. This makes it unnecessary to train m neural nets for each branching point. A huge improvement in speed is now realized. In real-world applications, fewer features means fewer costs to build sensors and to run systems. This paper also reports two successful real-world applications of the ANNIGMA-wrapper approach in helicopter maintenance.

II. The ANNIGMA-Wrapper Approach

A. An Approximate Metric for Total Gain

The heuristic called ANNIGMA ranks features by relevance based on the weights associated with the features. The reasoning behind this heuristic is that neural net weights can be viewed as representing the gain of the input signal to the output node. Input signals that are noisy or irrelevant to the output will have a high error rate if they have high associated weights. Therefore, training algorithms must reduce their weights such that they do not contribute to the output. In a similar manner, the weights of relevant and noise-free signals will be increased.

The equation for a two-layer neural net with the first layer having a logistic activation function, $\mathcal{S}(x) = \frac{1}{1+\exp(-x)}$, and the second layer having a linear output, is:

$$O_k = L_k \times \sum_j \mathcal{S}\left(\sum_i A_i \times W_{ij}\right) \times W_{jk},$$

where i, j, k are the input, hidden, and output layers node indices, respectively. L are the second layer linear multiplier values; A are the input nodes (features); O are the output nodes; and W are the weights between the layers. The output, O_k, as a function of a single input A_i, can be expressed as

$$O_k = L_k \times \sum_j \mathcal{S}\left(A_i \times W_{ij} + C_{ij}\right) \times W_{jk}, \qquad (1)$$

where C_{ij} represents the constant value of all the other inputs, including biases. C_{ij} here acts as a "setpoint" on the logistic function curve. This equation, with all of the inputs processed through the logistic function, is too complex to analyze directly. An approximation can be made of the total relative gain, G, of a particular input node i to a particular output node j. The approximation substitutes a linear factor for the logistic activation function. The approximation's error is reduced when the inputs are all in the same range. If we substitute a linear factor, F, for the activation function, we have

$$O_k \cong L_k \times \sum_j F \times (A_i \times W_{ij} + C_{ij}) \times W_{jk}. \quad (2)$$

The local gain, LG, is defined to be

$$LG_{ik} = \left| \frac{\Delta O_k}{\Delta A_i} \right|. \quad (3)$$

Since L_k and F are common factors to ANNIGMA's numerator and denominator, they can be dropped in the calculation of LG, i.e.

$$LG_{ik} = \sum_j |W_{ij} \times W_{jk}|. \quad (4)$$

The ANNIGMA score is the local gain, LG, normalized to a scale of 100:

$$\mathrm{ANNIGMA}_{ik} = \frac{LG_{ik}}{\max(LG_k)} \times 100. \quad (5)$$

When the input features are scaled in the same range, the weights will give an approximation of the relative gain of each of these features.

B. An Illustrative Example of the ANNIGMA Heuristic

We synthesized a small data set to test and illustrate the predictive power of ANNIGMA heuristic. The data set for this example has binary valued features that satisfy the normalization requirement of the ANNIGMA heuristic. The data set contains six columns of input features (A_1-A_6) and one column of target outputs (O_1). Table I shows sample records of our synthesized data.

Random	XOR		Sum < 2			Target
A_1	A_2	A_3	A_4	A_5	A_6	O_1
Rand(0,1)	1	1	1	1	1	0
Rand(0,1)	0	0	1	1	0	0
Rand(0,1)	1	0	0	0	1	1
Rand(0,1)	0	1	0	1	0	1
Rand(0,1)	0	1	1	0	0	1

TABLE I

SAMPLE RECORDS OF SYNTHESIZED DATA SET

Under uncorrupted conditions, the target output O_1 and the meta-signals XOR and SUM<2 must agree, and other combinations of features are not possible (i.e., $XOR(A_2, A_3) = (A_4 + A_5 + A_6 < 2) = O_1$). To simulate situations where noise exists along with redundancy, the fourth feature (A_4) was corrupted by flipping the bit with a twenty percent probability. (i.e., The value of A_4 is replaced by its complement with twenty percent probability (i.e., $P[A_4 \leftarrow not(A_4)] = 0.20$)). This tests the ability of the heuristic to select the noise-free features.

The ANNIGMA scores of these features after six iteration of training epochs is given in Figure 1. For each of the six features (feature $1-6$), there are six bars representing the ANNIGMA scores, G, for each of six neural net training iterations. The rightmost (black) bar of each feature reflects a fully trained net and is the final score. Figure 1 shows that within a few training iterations, ANNIGMA correctly suppressed the noisy inputs, A_1 and A_4. ANNIGMA correctly identified the XOR signal (A_2 and A_3) as the most relevant. We obtained the same results in thirty of thirty different neural net initializations.

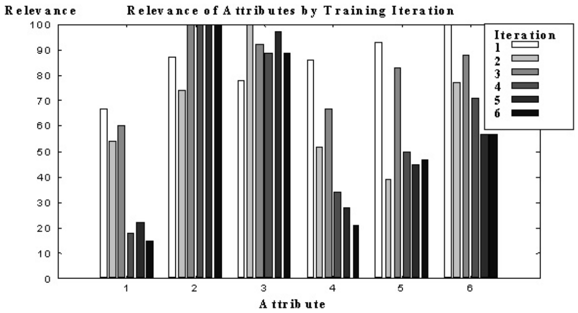

Fig. 1. ANNIGMA output by training iteration.

C. Integrating ANNIGMA into the Wrapper Algorithm

Selecting features solely based on a weight-based metric may yield unreliable results. The ANNIGMA-wrapper approach integrates the ANNIGMA heuristic into the wrapper model to achieve reliable results. Figure 2 gives an overview of the generic feature selection algorithm in the **ANNIGMA-wrapper** approach. The input to this algorithm is the set of all features, and their associated training data. The output of this algorithm is a feature subset.

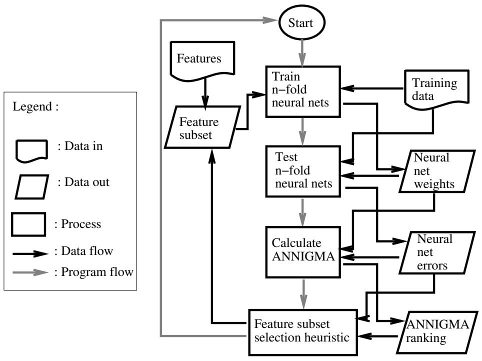

Fig. 2. Program and data flow in the ANNIGMA-wrapper algorithm

On the top level, this algorithm consists of two nested loops. The outer loop, called *the wrapper cycle*, selects the next subset to evaluate based on the ANNIGMMA evaluation of each feature and the classification performance of the neural network using this subset of features. How candidate feature subsets are actually generated and evaluated is realized by the box "feature subset selection heuristic" at the bottom of Figure 2. Feature subset selection heuristic is instantiated by a search strategy which will be described in detail in the next section.

The inner loop called *the training cycle* corresponds to the first and second boxes on the program flow path in Figure 2. Each of the training cycles takes the candidate feature set as the input and estimates the error rate of a trained neural net using the test data subset. The error rate is estimated by ten-fold cross-validation as follows. There is an additional "hold-out" set of data for final performance testing which is not used here.

The next box on the path of the program flow is to calculate ANNIGMA rankings of the features. After each run of the cross-validation is completed, Eq. 5 is evaluated for each feature as described in Section II-A. The resulting ANNIGMA score is then weighted by the test error rate (i.e., ANNIGMA_score * (1-test_error_rate)). This ensures that the better performing neural nets have proportionately greater influence on the final ranking of features. The resulting error-weighted ANNIGMA scores are averaged for n runs to produce the final score for each feature and the ANNIGMA ranking. We compared the results of weighted and non-weighted cases in our experiments. As expected, the weighted case performed better [5].

D. Search Strategies

Our algorithms are based on the strategie of backward elimination [1], [3]. Backward elimination starts with a complete set of original features and removes features from candidate subsets during the search. We present three versions of backward selection including greedy backward elimination (BE), backward elimination with backtracking (BEB), and backward stepwise elimination (BSE) for integrating the ANNIGMA heuristic into the wrapper model.

BE is a greedy version of backward elimination that runs the training cycle to obtain the ANNIGMA rankings and the error rates with all features, then starts with a set including all features. It repeatedly eliminates the next worst ANNIGMA ranked feature in each wrapper cycle until the error rate goes up.

BEB is a version of BE that allows for backtracking. The idea is that if the error rate goes up, instead of terminating the feature selection, the previous feature being eliminated is restored and the next worst ranked feature is eliminated. The process is iterated until a performance-improving elimination is found for each size of feature subsets.

BSE (Algorithm 1) is designed to accelerate feature selection for large data sets when BEB is too slow. The main idea is to eliminate a large number of seemingly irrelevant features in early cycles and adjust the feature subset carefully in the subsequent cycles. When the performance degrades, the best of the discarded features are brought back into the candidate subset.

Algorithm 1[Backward Stepwise Elimination (BSE)]

01. Run a training cycle with all feature, calculate ANNIGMA ranking $A(0)$
02. Let error(0) = averaged error rate and the feature subset $f(0)$ = all features
03. Let H = the set of discarded features, initially empty
04. Let cycle counter $t = 1$
05. WHILE NOT termination condition
06. IF $t < 4$ THEN LET $f(t)$ = top p_1% of the best features in $A(t-1)$
07. ELSE IF error$(t-1) \leq$ minimum of previous errors THEN
08. Let $f(t)$ = top p_2% of the best features in $A(t-1)$
09. ELSE IF error$(t-1) \leq$ mean of previous errors THEN
10. Let $f(t)$ = top p_3% of the best features in $A(t-1)$ plus the best feature in H
11. ELSE IF error$(t-1) \leq$ maximum of previous errors THEN
12. Let $f(t)$ = top p_4% of the best features in $A(t-1)$ plus the best 2 features in H
13. ELSE Let $f(t)$ = top p_5% of the best features in $A(t-1)$ plus the best $|f(t-1)/2|$ features in H
14. Update H; Sort H over the history-averaged ANNIGMA scores
15. Run a training cycle with $f(t)$, calculate ANNIGMA ranking $A(t)$
16. $t = t + 1$
17 END WHILE and RETURN $f(t)$

The parameters ($p_1 \sim p_5$) used in Algorithm 1 are tuned for the experiment in the pulse-echo classification domain ($p_1 = 75$, $p_2 = 85$, $p_3 = p_4 = 90$, and $p_5 = 50$; see Section V). The reduction percentages (lines 6, 8, 10, 12, and 13) and initial fixed reduction period (line 6) can be adjusted for a given application. Changes to these parameters may affect the efficiency of the search as well as the selected features. In our experiments, we found that slight changes to these parameters did not affect the resulting sets of selected features.

History-averaged score is used as a more reliable estimate of the relevance of a discarded feature than the most recent score because re-introducing a discarded feature is based on an assumption that its relevance is underestimated by the most recent score. However, which score in the previous wrapper cycles is the most reliable is not known, while re-computing the score may incur huge overhead. Therefore, a conservative choice is to use the history-averaged score.

III. Experimental Results Against UCI Data sets

This section reports experiments exploring the results of using the ANNIGMA-wrapper algorithm against standard artificial and real-world data sets. These data sets were obtained from the UCI Machine-Learning Repository [6].

A. Data Set Description and Preparation

The first column of Table II gives the size of our experimental data sets (in the format: training/cross-validation + hold-out set.) The ratio of training and hold-out set is two to one unless explicitly designated by data set providers. Some data sets contain missing values. We simply replace all missing values by the feature mean for all instances belonging to the same class. The first four rows are artificial data sets and the rest are real-world data sets.

B. Experimental Procedure

The second column of Table II describes the neural net configurations (in the format: maximum epochs + number of hidden nodes + hidden layer transfer function + output layer transfer function) for each data set in the experiments. These configurations are empirically determined based on their cross-validation errors and relative variances of the resulting ANNIGMA rankings.

The experiments are carried out as follows. For each data set, the first step is to normalize input features into the same range, usually between 0 and 1. After the configuration is determined, we estimate the error rate of the neural net with no feature selection by applying 10-fold cross-validation to train 10 sets of the weights using the

data set	data size	**configuration**	Algorithm
3P	100+40	8+6+logsig+purelin	BEB
CorrAL	64+64	10+8+tansig+purelin	BE
Monk3a	122+432	15+6+tansig+purelin	BE
Monk3b	122+432	8+3+purelin+purelin	BE
Cancer	399+300	10+12+logsig+logsig	BEB
Credit	490+200	10+10+tansig+purelin	BE
Heart(LB)	133+67	10+2+purelin+purelin	BEB
Ionosphere	200+151	10+22+logsig+purelin	BSE
Pima	576+192	10+6+tansig+logsig	BEB
Vote	218+217	8+3+purelin+purelin	BE

TABLE II
INFORMATION FOR EACH DATA SET

training set, and then testing them against the hold-out set and averaging the resulting error rates. Next, we compare the feature selection performance of ANNIGMA-wrapper by applying different search strategies described in Section II-D thirty times and report the average number of features selected, the average error rate, and the average execution time. In each trial, the elements in training set and hold-out set of a testing data set are randomly selected except the data set whose training and hold-out set have been explicitly designated by its provider. The error rate is estimated by averaging hold-out set errors after the final feature subset is selected. Two different search strategies are applied to each data set for performance comparisons: "SWB", and "BE family". "SWB" is the standard wrapper-backward elimination. It is a greedy version of backward elimination for the standard wrapper feature selection, where error rates are the sole metric to guide the search. This approach is introduced here for the purpose of comparisons. The fourth column of Table II shows which "BE family" algorithm is used for each data set. We use BSE for data sets with many features, BE for small or synthesized data sets, and BEB for others, mainly to maximize the utility of our computing facilities. The experiments are conducted on Pentium III 800MHz PCs with 128MB memory.

C. The Results

Table III reports the average execution time in seconds to complete a feature selection task for each data set with different algorithms. In all cases, the ANNIGMA-wrapper algorithms (BE family) can complete a feature selection task in less than ten minutes. Compared with SWB, the ANNIGMA-wrapper algorithms are many times faster for all data sets, especially for those data sets with a large number of features, such as Ionosphere (34 features).

Table III also reports the feature selection performance of the ANNIGMA-wrapper approach. The "NN" column lists the results with no feature subset selection. The second column ("SWB") reports the results of the standard wrapper-backward elimination. The third column compare the results by using the search strategies (BE family) of the ANNIGMA-wrapper approach. For each case (except for SWB for Ionosphere), we report the average and the standard deviation of the number of features selected, the average and the standard deviation of the error rates. Since it takes too much time for SWB to select features for data set Ionosphere, we only complete this algorithm one time for Ionosphere and report that only result.

We also use ANOVA with the Bonferroni procedure for multiple comparisons statistics [7]. The difference between any two error rates in a row must be at least as large as the value in the *RD* (*Required Difference*) column in order to be considered statistically significant at the 90% confidence level for the experiment as a whole. An error rate in **boldface** is significantly better than that of "NN".

D. Discussion

The results show that BE family perform well in general for seven out of the ten data sets, improving the number of features or the error rate over the base NN significantly. The results for 3P and CorrAL are particularly remarkable. BEB selects the correct three features for 3P and achieves perfect classification in all 30 tests. Even the replicated features are distinguished and eliminated. For CorrAL, BE almost always selects the correct four features, occasionally adding an irrelevant one.

The results also show that feature expansion can have a negative effect when it does not offset the cost of adding another feature. For Monk3a, using features as-is, the correct features were selected. For Monk3b, the correct features were chosen, but never the minimum subset. Note that when using the expanded representation, there are aliases for each feature (e.g., a4 = 1 implies that a4 $\neq$ 2 and a4 $\neq$ 3). Comparing the results for Monk3a with Monk3b, we see that ANNIGMA-wrapper can identify relevant features even when they are multiple-valued. We also tried to perform feature expansion for some data sets (such as Credit) which have non-binary categorical features, but results, not reported here, were time-consuming and the perforamces are not improved significantly.

IV. COMPARISONS WITH RECENT RELATED WORK

Recently, many clever techniques have been proposed to improve the effectiveness of feature selection for neural nets, ranging from filter model-based approaches to genetic algorithms. This section compares their results with ours.

A. Recent Advances

Richeldi and Lanzi presented an approach called AHOC [8], which partitions the observed features into a number of groups, called factors, that reflect the major dimensions of the phenomenon under consideration. A genetic algorithm is used to explore the feature space originated by the factors and to determine the set of the most informative feature configurations.

Neural Network Feature Selector (NNFS) [9] is a method that adds a penalty term to the error function used to derive the weight updating rule of neural network training.

GADistAl [10] is a wrapper-based approach to feature selection using a genetic algorithm in conjunction with a constructive neural network learning algorithm called DistAl [11], which is employed to evaluate the "fitness" of candidate feature subsets in the genetic algorithm. The

data set	NN		SWB			BE family			RD	Recent advances		
	F.	Error%	F.	Error%	Time	F.	Error%	Time		F.	Error%	Ref.
3P	13	9.3±14.7	4.4±3.1	**0.5±1.4**	1165	3.0±0.0	**0.0±0.0**	175	5.3	3.0±0.0	0.0±0.0	hybrid
CorrAL	6	2.3±4.0	4.0±0.2	**0.0±0.2**	98	4.0±0.2	**0.0±0.2**	50	1.4	4.0±0.0	12.5±0.0	INFO
Monk3a	6	10.0±5.2	3.4±1.6	**5.1±3.4**	125	2.3±0.7	**2.9±0.8**	58	2.2	–	–	–
Monk3b	15	2.8±0.0	4.4±1.1	2.8±0.0	519	2.2±0.4	2.8±0.0	56	0.0	3.9±1.8	1.6±1.7	NNFS
Cancer	9	4.1±4.7	7.2±1.2	3.6±1.1	451	5.8±1.3	3.5±1.2	280	1.8	2.7±1.0	5.9±1.0	NNFS
Credit	15	14.1±1.7	13.4±1.0	14.4±0.8	1712	6.7±2.5	**12.0±0.8**	375	0.7	7.0±0.0	14.9±6.1	AHOC
Heart(LB)	13	20.2±0.2	7.3±1.9	25.7±1.8	352	2.7±1.2	22.3±2.0	121	1.0	5.0±0.0	19.2±6.5	AHOC
Ionosphere	34	11.4±3.9	32	10.2	138065	9.0±2.5	**9.8±1.3**	342	1.3	–	–	–
Pima	8	24.1±5.0	6.9±1.0	23.0±1.3	179	5.2±1.4	**22.2±1.4**	168	1.9	2.9±0.2	25.7±3.3	NNFS
Vote	16	3.2±0.0	3.2±1.4	3.2±0.2	1052	3.3±1.9	**3.1±0.2**	65	0.1	5.0±0.0	4.3±3.5	AHOC

TABLE III

Performance Statistics of feature selection by ANNIGMA-wrapper

"fitness" function is designed to combine the classification accuracy and the cost of using a set of features.

Dash and Liu [12] proposed a hybrid algorithm of probabilistic search [13] and complete search to take advantage of both algorithms. It begins with Las Vegas filter(LVF) [4] , a probabilistic feature selection algorithm, to reduce the number of features and then runs "Automatic Branch & Bound (ABB)", a complete search algorithm.

B. *Comparisons*

The algorithms that are surveyed for the comparison including the original wrapper [1], Las Vegas filter(LVF) [13], Las Vegas wrapper(LVW) [4], neural net feature selector(NNFS) [9], hybrid approach(hybrid) [12], information-theoretic filter(INFO) [14], and AHOC genetic algorithm (AHOC) [8][1]. The last column of Table III shows these performance data. The comparison is made on the basis of both the number of features selected and the error rates after feature selection. It should be mentioned that results achieved by different algorithms are not obtained with the same experimental procedure, nor did we implement and rerun their algorithms but using their report literally. Therefore, the comparison is inherently informal.

According to the performance data, among 8 data sets, ANNIGMA-wrapper is better in 6 cases (3P, CorrAL, Cancer, Credit, Pima and Vote) in terms of the error rates. Hybrid [12] also achieves perfect result for 3P, but their 3P data set contains 12 features while we use a 3P data set with 13 features. For data sets Monk3b, and Heart(LB), ANNIGMA-wrapper produces the smallest feature subset but higher error rates than the best of recent advances. However, we note that the training data set of Monk3 has 5% class noise, and the hold-out data set contains the training data set. The 12 noisy records out of a total of 432 records result in an expected best error rate of 2.78%. This is exactly the error rate achieved by ANNIGMA-wrapper in most of its cases. An induction algorithm achieving a better error rate might *overfits* the data by modeling the noise.

[1]Table III does not include the results of GADistAl because they reported 10-fold cross validation results rather than the results for hold-out data. Their results may not be comparable with other approaches.

V. Applications in the Helicopter Industry

This section reports two successful applications of the ANNIGMA-wrapper approach in the helicopter industry.

A. *Pulse-Echo Experiment*

This system classifies ultrasonic pulses that are used to inspect parts formed of laminated composite materials. In this system, the inspection process consists of sending a short ultrasonic pulse into the material under the test. These echo signals are collected and digitized by an A/D converter. Any change in the materials propagation speed for sound within the sample will result in an echo being sent back toward the transducer that initiates the test pulse. Material failures are associated with echo signal shapes.

The digitized waveform is converted into 192 features. The system must classify these digitized waveforms into "good" and "bad" categories. An existing system uses a wavelet-compression approach to select top 25 features.

In this experiment, the ANNIGMA-wrapper algorithm uses all 192 features as its initial feature set. One cycle takes about one minute on a Pentium-Pro 200MHz equipped computer. Figure 3(a) shows the features selected as the ANNIGMA-wrapper algorithm progressed for material "45b45". Figure 3(b) shows the number of features selected by a wrapper cycle. The associated performance is presented in Figure 3(c). The accuracies are based on 150 pulse-echo waveform that were not used in the training. When this ANNIGMA-wrapper algorithm was run, stopping criteria was turned off. It can be seen that cycles after cycle 48 are not necessary.

The existing system, using the top 25 wavelet coefficients, achieved 97% accuracy. The ANNIGMA-wrapper algorithm, used only 10 of these coefficients to achieve 99% accuracy. The intersection of the two sets of coefficients contains only two members. The result suggests that the set of wavelet coefficients contains a large number of redundant relevant features. The ANNIGMA-wrapper approach manages to select a smaller set of features while achieving a slightly better accuracy.

B. *Strain Signal Prediction*

The second application is the development of a system that predicts the strain on helicopter rotor blades. Helicopter rotor blade repair and replacement are major cost

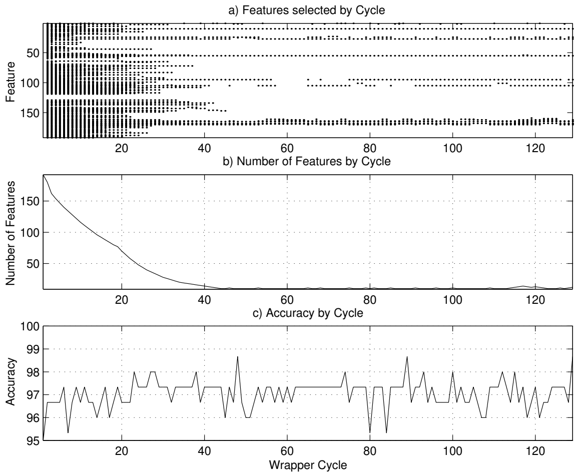

Fig. 3. ANNIGMA-wrapper results by cycle for material 45b45

factors in helicopter operations. However, they are essential to assure helicopter safety. Current industry standard to measure the fatigue life is based on a fixed number of operating hours assuming most severe operation. This is an inefficient because perfect blades may be mistakenly scrapped. A proposed solution is to track the strains on the blades to calculate fatigue damage [15] [16]. However, it is not feasible to directly measure strains on production helicopters due to the high expense of a sensor/monitoring system, and the unreliability of a strain sensor mounted on a rotor blade. We attempt to solve this problem through a neural net trained to predict strain from other easily obtainable helicopter signals. The neural nets in this application is to approximate the strain gauge reading mounted on the helicopter blades as a function of the given sensory inputs. Initially, forty-one sensors from among thousands available on the helicopter were selected by flight dynamics experts. We have 197 sets of training datasets (1G Bytes large total). Each set contains the sensory data collected from 3 to 19 seconds periods.

Due to very large size of the data sets, the first wrapper cycle takes 200 hours of exclusive run-time on a 200 MHz Pentium Pro-based computer. After 3 wrapper cycles, ANNIGMA-Wrapper selects 19 features from 41 and reduce the error rate from 17.3% to 16.3%. The fourth cycle yields 13 features but the error rate jumps to 77.4%. The implication of these results is that the information content in the 13 selected features is insufficient compared to that in the 19 features. These 19 signals are therefore recommended for further system development. The first wrapper cycle was re-run to check the reliability of the ANNIGMA heuristic. We found that the top 10 high-ranking features were within two ranks of differences between two runs [5].

VI. Conclusions

In this paper, we have presented a new approach to selecting features for neural nets called ANNIGMA-wrapper that makes the wrapper model feature selection tractable in real-world neural net applications. Experimental results against standard data sets from UCI repository show that our simple approach performs well for data sets with various characteristics, and the ANNIGMA-wrapper approach was successfully applied to two real-world neural net applications in the helicopter industry.

Acknowledgements

We wish to thank Mrs. Ya-Ting Yang for her help in the programming our experiments and anonymous reviewers for their valuable comments.

References

[1] G.H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," in *Machine Learning : Proceedings of the Eleventh International Conference (ICML '94)*, San Francisco, CA, 1994, Morgan Kaufmann.

[2] R. Kohavi and D. Sommerfield, "Feature subset selection using the wrapper method: Overfitting and dynamic search space topology," in *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD '95)*, Menlo Park, CA, 1995, AAAI Press.

[3] R. Carunan and D. Freitag, "Greedy attribute selection," in *Machine Learning : Proceedings of the Eleventh International Conference*, San Francisco, CA, 1994, Morgan Kaufmann.

[4] H. Liu and R. Setiono, "Feature selection and classification - a probabilistic wrapper approach," in *Proceedings of the 9th International Conference on Industrial and Engineering Applications of AI and ES*, 1996.

[5] Dietrich Schuschel, "Automation of attribute selection for neural nets," Thesis for Master of Science in Computer Science, Department of Computer Science and Engineering, Arizona State University, 1998.

[6] C. J. Merz and P. M. Murphy, "UCI Repository of machine learning databases," 1998, `http://www.ics.uci.edu/~mlearn/MLRepository.html` Irvine, CA: University of California, Department of Computer Science.

[7] J. T. McClave and F. H. Dietrich, *Statistics*, Dellen Publishing Company, San Francisco, 1991.

[8] Macro Richeldi and Pier Luca Lanzi, "Performing effective feature selection by investigating the deep structure of the data," in *Proceedings of the Second International Conference on Knowlwdge Discovery and Data Mining (KDD '96)*, E. Simoudis, J. Han, and U. M. Fayyad, Eds. 1996, AAAI Press.

[9] R. Setiono and H. Liu, "Neural network feature selector," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, 1997.

[10] Jihoon Yang and Vasant Honavar, "Feature subset selection using a genetic algorithm," in *Feature Extraction, Construction, and Subset Selection: A Data Mining Perspective*, H. Motoda and H. Liu, Eds., chapter 8. New York: Kluwer, 1998.

[11] J. Yang, R. Parekh, and V. Honavar, "DistAL: An interpattern distance-based constructive learning algorithm," *Intelligent Data Analysis*, vol. 3, pp. 55–73, 1999.

[12] Manoranjan Dash and Huan Liu, "Hybrid search of feature subsets," in *PRICAI'98*, Singapore, November 1998.

[13] H. Liu and R. Setiono, "A probabilistic approach to feature selection - a filter solution," in *Proceedings of the 13th International Conference on Machine Learning (ICML'96)*, Bari, Italy, 1996, pp. 319–327.

[14] Daphne Koller and Mehran Sahami, "Toward optimal feature selection," in *Machine Learning: Proc. of the Thirteenth International Conference (ICML '96)*, Lorenza Saitta, Ed., San Francisco, 1996, CA:Morgan Kaufmann.

[15] S. Downing and D. Socie, "Simple rainflow counting algorithms," *International Journal on Fatigue*, January 1982.

[16] E ASTM 1049-85, "Standard practices for cycle counting in fatigue analysis," *Americam Society for Testing and Materials*, 1985.