Spam Email Classification using Artificial Neural Network with Shared Weights Auto-associative Neural Networks and Evolutionary Algorithm

Yue Wang

Reseach School of Computer Science, Australian National University,
2601 Canberra, AUSTRALIA
u5977335@anu.edu.au

Abstract. Spam email is a major problem of the contemporary Internet. This paper proposes three artificial neural network based models for spam email classification. The first model is a multilayered artificial neural network (ANN) with backpropagation. The second two models are based on the first model with an additional auto-associative neural network for feature extraction (AANN-ANN) and evolutionary algorithm for feature selection (EA-ANN). For basic ANN model, I use accuracy, F-measure and testing time to evaluate. For the second two models, except three measures used in ANN model, training time and compressed or selected features size are also used to compare two dimensionality reduction algorithms in the models. Several comparisons are organised between three models from different aspects. The results show that AANN-ANN model has the highest accuracy (0.90), F-measure (0.86) and the smallest FP number (156). Its testing time is less than ANN but a little longer than EA-ANN. However, its training time is significantly shorter than EA-ANN. Adding feature reduction algorithm consumes more time to train the model, but it can reduce the storage space for data after the extraction or selection and reduce the testing time. Finally, three models in this paper are compared with other four approaches based on machine learning algorithms presented by Hidalgo, Lopez and Sanz on the same dataset. My three models are at the same level of performance ranking of seven approaches. For accuracy, My models are approximately 0.89, better than Naive Bayes algorithms and 5 Nearest Neighbours but worse than C4.5 and Part. For F-measure, all my models have higher F-measure (about 0.85) thus they are more robust than others.

Keywords: Artificial neural network; Feature extraction; Shared weight auto-associative neural network; Feature selection; Evolutionary algorithm.

1 Introduction

Email is an important part of modern life. It is a fast and economical way to communicate with people all over the world as long as we know their email addresses. However, this trait makes lots of unsolicited bulk emails come into our e-mail box. These emails are known as spam email, like the advertisement, violence information, viruses and so on. Dealing with spam emails wastes time and some content can be harmful to users. Therefore, approaches which can identify spam emails is essential.

Since the spam emails have started to grow, many approaches have been proposed to distinguish spam emails and block them. In paper [1], Erosheva and Fienberg apply a fully Bayesian approach to soft clustering and classification using mixed membership models. Tretyakov evaluates the performance some of the most popular machine learning methods including Bayesian classification, k nearest neighbour (kNN), artificial neural network and support vector machines (SVMs) for spam-filtering and they also make experiments on combined classifiers of two methods [2]. A hybrid approach combines Collaborative Filtering (CF) and Content-Based Filtering (CBF) is proposed by Cortez, Lopes and Sousa, et al., called Symbiotic Data Mining [3]. This approach is more competitive than CBF and more robust to contamination attacks. Yang and Elfayoumy use the genetic algorithm to train a multi-layer perceptron (MLP) and then use the trained MLP to do spam filtering [4]. This filtering system achieves an accuracy of about 94% to detect spam emails, and 89% to detect legitimate emails.

In this paper, I firstly train a multi-layered artificial neural network (ANN) to classify spam emails and regular emails from UCI spambase dataset [5]. However, too many features result in slow computation and may contain some redundancy or irrelevant features which can worsen the results. One method is to reduce the dimensionality of features. I modify the basic model by two approaches. The first approach, called AANN-ANN, is applying an auto-associative neural network (AANN) to implement feature extraction before training and testing ANN classifier. During AANN training, shared weight constraint [13] is conducted. Another approach is using the evolutionary algorithm (EA) to make feature selection, then train and test ANN classifier, which is called EA-ANN approach. To reduce the computation in fitness calculation in EA, the idea of using a pre-trained neural network and masked input features to give evaluations of the selected subset of features [14] is applied. All approaches are measured by accuracy, F-measure, false positive number and testing time. For AANN-ANN and EA-ANN, the training time and compressed features size are also recorded to measure two features dimensionality reduction algorithms. Several comparisons are organized among three models from different aspects. Then, ANN, AANN-ANN and EA-ANN are compared with Hidalgo, Lopez and Sanz's work [6], which evaluates four machine learning algorithms for classifying the same dataset. The results show, among my three models,

AANN-ANN has the best performance with the highest accuracy (0.90), the highest F1-measure (0.86) and the smallest false positive number (156). Its testing time is also acceptable, shorter than ANN but a little longer than EA-ANN. However, its training time is significantly shorter than EA-ANN. For the comparison with algorithms in [6], my models are at the same performance level. They are more robust than algorithms in [6] because they have higher F-measures (between 0.85 to 0.86). Their accuracies (between 0.89 to 0.90) are higher than Naïve Bayes algorithm and 5-Nearest Neighbours but less than C4.5 and Part.

2 Method

In this section. I first describe the dataset and the method for pre-processing data (section 2.1 & 2.2). Then, I propose an ANN model to classify spam and non-spam emails (section 2.3). Next, I modify the ANN model in two ways, by adding AANN for feature extraction (section 2.4) and by adding EA for feature selection (section 2.5). Finally, I introduce the evaluation measures (section 2.6).

2.1 Dataset

The dataset used in this paper is spambase from UCI machine learning repository created by Dua and Karra Taniskidou [5]. It contains 4601 instances. The numbers of positive instances (spam) and negative instances (non-spam) are 1813 and 2788 respectively, which is not very distorted and suitable for training. Each instance has 57 attributes and 1 label. Attributes description is in Table 1.

Table 1. The attributes description.

Attributes No.	Name	Type	Description
1-48	word_freq_WORD	continuous real [0,100] number	percentage of the specific word in an email
49-54	char_freq_CHAR	continuous real [0,100] number	percentage of the specific character in the email
55	capital_run_length_average	continuous real [1,...] number	the average length of uninterrupted sequences of capital letters
56	capital_run_length_longest	continuous integer [1,...] number	length of longest uninterrupted sequence of capital letters
57	capital_run_length_total	continuous integer [1,...] number	total number of capital letters in the email

2.2 Data preprocessing

In general, normalisation can improve convergence speed of the gradient descent [7] and the accuracy of the model. In this paper, the data are normalised to the range [0, 1] using the feature scaling. Denote x in attribute A is normalised to x^{new}. min is the minimal value and max is the maximal value in attribute A. We have the formula (1) to calculate x^{new}.

$$x^{new} = \frac{x - min}{max - min} \quad (1)$$

2.3 Proposed multi-layered ANN model

The proposed ANN model has one input layer, two hidden layers and one output layer. The input layer has 58 units, equal to the number of features plus one bias. Both hidden layers have 120 units and 1 bias unit. The output layer has 2 units, representing two classes. The structure is shown in Fig. 1.

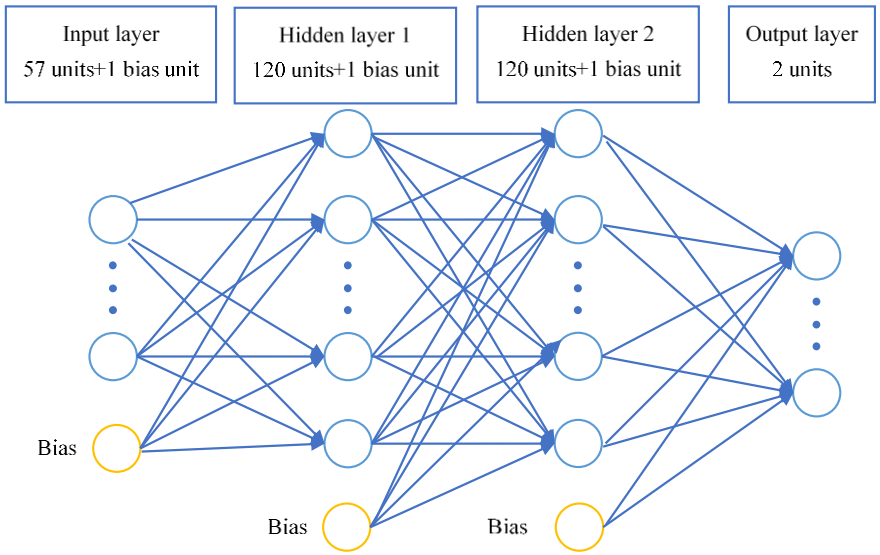

Fig. 1. ANN classifier structure, including one input layer, two hidden layers and one output layer.

The activation function is the sigmoid function. The softmax function is used to project output into (0, 1) so that it can represent the probability of current instance being classified in each class. Denote the output is a vector *z* of length *K*, the softmax function is defined as

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad for\ j = 1, \dots, K \tag{2}$$

The class *j* which has the highest probability is the label model predict for this instance. The loss function used here is cross entropy (3).

$$\mathrm{E(w)} = -\ln \mathrm{p(t|w)} = -\sum_{n=1}^{N} \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\} \tag{3}$$

2.5 ANN with Shared weight auto-associative neural network for feature extraction

Auto-associative neural network (AANN) is a type of network that can compress the inputs to a small sample and restore the small sample to original inputs. The number of input units and output units are equal. And one layer between the input layer and output layer should have smaller size compared with the input layer and output layer. We can regard it as a process of encoding and decoding. AANN is widely used in dimensionality reduction [10], filtering [11], classification [12], etc.

Since the spambase dataset has many zero values, it may contain redundant features. If the data contain many redundant or irrelevant features, we can select or extract a small set of features without resulting in much loss of information [8]. Meanwhile, feature selection or extraction can simply the model and make it easy to understand [9], shorten the training time and strengthen generalisation by reducing overfitting [8]. Thus, in this paper, I use AANN to do feature extraction, which reduces the dimension of features by creating new features from the original features.

To improve the extraction quality of AANN, I put shared weight constraint proposed by Gedeon, Catalan and Jin [13] on AANN. The topology of shared weight AANN is symmetrical. The weights of the corresponding positions of the right part of net and left part net are same.

Following Gedeon's idea, the AANN I designed for feature extraction is shown in Fig. 2. It contains one input layer (57 units), one hidden layer (less than 57 units) and one output layer (57 units). The number of units in hidden layer *n* is controlled by the compression ratio (4).

$$n = floor(the\ number\ of\ input\ units * compression\ ratio) \tag{4}$$

The connections between units are simple linear transformation. No activation function used in units. The outputs of hidden layer are the extracted features and will be used as inputs for ANN classifier. Training data and targets are both the 57 attributes since the output layer should restore the original input. The loss function I used here is the mean squared error between *n* elements in the input x and target y. The loss function can be described as (5):

$$\mathrm{MSE} = \frac{1}{N} \sum_{n=1}^{N} (x_n - t_n)^2 \tag{5}$$

The advantage of weight sharing is that it constrains the weights and reduces the number of free parameters. In this condition, we only need to decide half of the parameters compared with the standard AANN. The short of this method is

that the outputs of hidden layer could only restore an approximation of the original inputs because this network is inherently lossy.

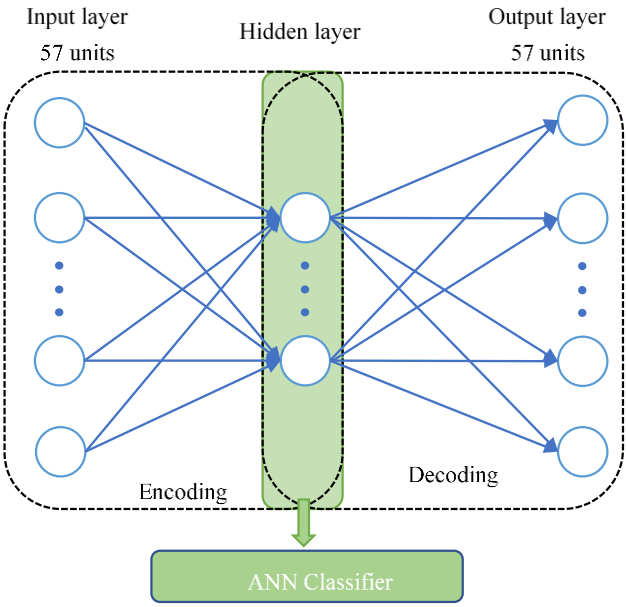

Fig. 2. The structure of AANN used for feature extraction and how it connects to the ANN classifier.

2.6 ANN with evolutionary algorithm for feature selection

Evolutionary algorithm (EA) is a generic population-based metaheuristic optimisation algorithm. EA is easy to develop and test. If the problem-specific algorithm is unknown or expensive, the EA can give a good solution in acceptable time. In this paper, I use EA to do feature selection, which has the same purpose with feature extraction but should be distinguished from feature extraction. Feature selection selects a subset of relevant features and does not create new features.

A model combined ANN and EA algorithm for feature selection (EA-ANN) is proposed. The process of EA used for feature selection is shown in Fig. 3. Once EA algorithm gives out the best individual, we use the features represented by that individual to train a new ANN classifier. A detailed explanation of each step in EA is given in the following.

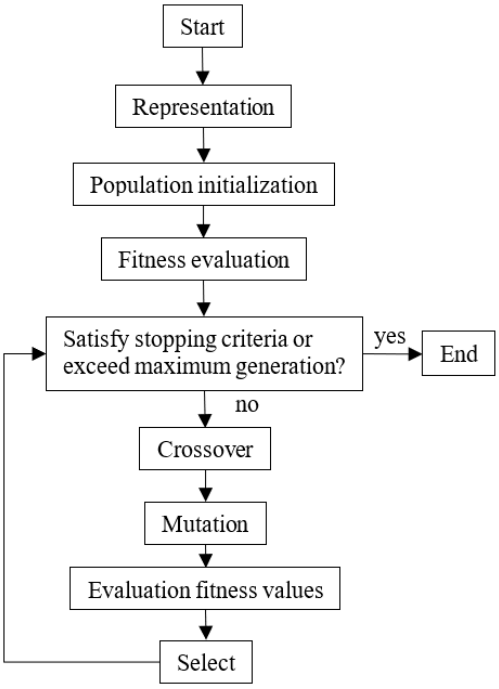

Fig. 3. EA algorithm for feature selection

Representation. The representation for a subset of features is encoded as a binary mask vector, regarded as an individual in this algorithm. The length of a mask vector is equal to the initial features dimension, each bit represents a feature. If the bit is 0, the corresponding feature is not selected. On the contrary, the corresponding feature is selected.
Population initialisation. Denote population size is P. First, I randomly generate P number from the uniform distribution [1, *the number of initial features*] and make them in a one-dimension vector called S, each value in this vector represents the size of a subset. Then P subsets are generated, the subset is represented by the binary mask vector as described before. The generation method is randomly selecting S_i positions to be 1 while others are 0 in i^{th} binary mask vector. Use this method instead of directly generating mask vectors by randomly selecting 1 or 0 for each bit is to ensure a uniform representation of entire search space is initialised. If I just randomly selecting 1 or 0 for each bit in each individual, the initial individuals will represent the subsets with similar size.
Fitness evaluation. The fitness function is used to evaluate the ability of an individual to survive. Higher fitness score means the individual has a stronger ability to survival to next generation. Fitness function employed in this approach is

$$\text{Fitness} = \text{F measure} + m * (1 - \text{NumOf(features in subset)}/\text{NumOf(all features)}) \quad \textbf{(6)}$$

where m is the parameter controls the number of features after selection. This fitness function considers F-measure and reduction degree together to evaluate individuals in a comprehensive way. To get F-measure for an individual, generally we need to use the subset of features represented by current individual from train set to train a new neural network model and test it [15]. However, this approach is very expensive in computation. Since my basic ANN classifier needs to train with large epochs, it is unpractical to use this approach in this situation. One idea to calculate it in an efficient way is proposed by Li [14]. Firstly, train a neural network with whole features set (Fig. 4). In this paper, I use the ANN model described in section 2.3. To test a new individual v, if v_i is 0, we mask the corresponding feature in input pattern by setting it to 0. If v_i is 1, the corresponding feature in input pattern keeps its original value. The masked pattern is fed to trained ANN model, ANN can give a prediction of this pattern. This process is illustrated in Fig. 5. Process all patterns in test set in this way we can get the F-measure. The result can give us a reasonable evaluation on how this selected subset contributes to the correct classification. More important, it is much faster and more practical for large basic ANN model. One thing needs to emphasize is that this model is not the final model for classification in this approach. It is just a method to calculate fitness effectively. After EA algorithm, the selected features will be used to train a new model.

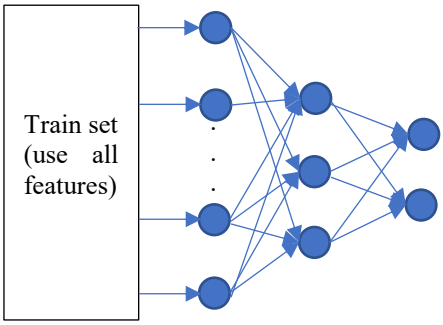

Fig. 4. Train an ANN with full features

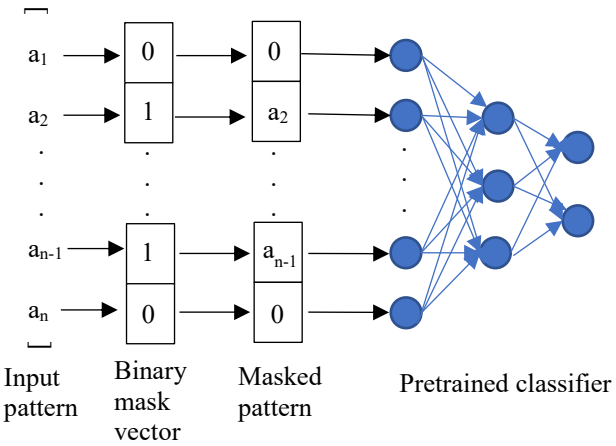

Fig. 5. Mask the input pattern and feed it to trained ANN

Stopping criteria. The algorithm stops when the generation exceeds maximum generation number or the best fitness in population does not change for ten generations.
Crossover and mutation. Uniform crossover and random mutation are used in this algorithm.

Selection. Selection is used in selection of parents and selection of offspring. For selection of parents, I select *P* (population size) parents called *Parent1* by proportional selection, and *P* parents called *Parent2* by random selection from population. Then do crossover between *Parent1* and *Parent2*. Each pair of parents can generate two offspring. I select the one with higher fitness to substitute *Parent1*. When *Parent1* is totally substituted by offspring, *Parent1* becomes the next generation.

2.7 Evaluation measures

The evaluation measures I use are accuracy, F-measure, false positive (FP) number and testing time. There is a relationship between accuracy and F-measure, but each one has a different emphasis. Accuracy is the most basic metric of how many instances are classified correctly. F-measure is a balance of recall and precision. Recall indicates whether the model is comprehensive, that means whether all spam emails are detected. Precision illustrates whether the model is precise, that means the detected spam emails are real spam emails. FP number is a sensitive indicator for email classification problem because people would rather see some spam than miss any important email. A good model should have low FP to minimise the damage. For testing time, it measures the time for predicting the test set. In this paper, the data set is split into 5 groups for 5-fold cross-validation. Thus there are about 920 patterns in the test set. For AANN-ANN model and EA-ANN model, I also use the training time, compressed or selected features size for dimensionality reduction algorithms (AANN and EA) to measure their performance.

The confusion matrix is calculated first because all these measures are based on confusion matrix. In this paper, the confusion matrix is

$$\begin{bmatrix} True\ Negative & False\ Positive \\ False\ Negative & True\ Positive \end{bmatrix}$$

True Negative (TN): The number of non-spam emails is classified as non-spam emails.
True Positive (TP): The number of spam emails is classified as spam emails.
False Positive (FP): The number of non-spam emails is wrongly classified as spam emails.
False Negative (FN): The number of spam emails is wrongly classified as non-spam emails.

As I use 5-fold cross-validation, the final output confusion matrix is the sum of confusion matrix of 5 times testing of 5 models. This does not influence the calculation of measures. Measures can be defined based on confusion matrix as

$$\text{Accuracy} = \frac{TN + TP}{TP + FN + FP + TN}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{F} - \text{measure} = \frac{2 * Recall * Precision}{Recall + Precision}$$

3 Results and discussion

3.1 Training methods

Before training and testing models, firstly Preprocess the data using the feature scaling normalisation as described in section 2.2 to get the normalised data. Then, we can train the models.

ANN model. Train ANN model with the 5-fold cross-validation. Spilt the whole dataset into 5 subsets by StratifiedKFold function from sklearn which can do stratified sampling. In each fold, train the ANN with the epoch = 1000, learning rate = 0.005 and batch size = 30.

AANN-ANN model. Split dataset like the partition in ANN model for cross-validation. In each fold, there are two phases of training AANN-ANN model.

(1) Train the shared weight AANN with the parameters epoch = 500, learn rate = 0.005 and minibatch methodology is used with batch size = 100. The outputs of the middle layer of AANN in last epoch are fed to ANN classifier to train ANN as the second phase.

(2) Train the ANN with the epoch = 1000, learning rate = 0.005 and batch size = 30. The number of input layer neurons should be consistent to the number of neurons in middle layer of AANN. Other layers are same with ANN model in section 2.3.

EA-ANN model. There are three phases of training EA-ANN model.

(1) Split dataset into 80% train data and 20% test data in a stratified fashion. Use the train data to train the basic ANN model with all features. This ANN is used to calculate fitness.

(2) Train EA with population size = 50, maximum generation = 100. Each parent has 0.8 probability to crossover with others. Each bit in offspring has 0.002 chance to mutate. The fitness function uses the test set spilt in the first phase to give F-measure. The EA algorithm stops when generation exceeds the 100 or the best fitness in population does not change for 10 generations. Then, the features which are represented by 1 in the best individual are selected.

(3) Split original dataset like the partition in ANN model for cross-validation. Train a new ANN with features selected by EA. The number of input layer neurons should be consistent with the number of neurons in middle layer of AANN. Other layers are same with ANN model in section 2.3.

3.2 Experiments and results

Experiment 1. Train and test ANN model. The motivation for this experiment is to get the evaluation results of ANN model as the baseline to compare with the following two modified models.

Results. The results of ANN model are shown in Table 2. The values are average values of 5 times running by 5-fold cross-validation.

Table 2. The results of evaluation of the ANN model.

	Accuracy	F-measure	FP number	Testing time
ANN	0.89	0.85	178	0.00738

Experiment 2. For AANN-ANN model, I designed 9 experiments with different compression ratios of AANN. The motivation is to explore how the performance of AANN-ANN model varies with different compression ratios. Record evaluation results for each experiment to compare with ANN model and EA-ANN model. We can select the best performed AANN-ANN to compare with baseline, ANN model, and the best performed EA-ANN model from different aspects, to figure out which model is the most suitable for this problem. Moreover, the overall trends of two modified models is also worth to compare to find some traits of two different dimensionality algorithms.

Results. The average results after 5-fold cross-validation are shown in Table 3. One thing needs to notice is that the testing time includes the time for features extraction and time for prediction.

Table 3. The results of evaluation of the AANN-ANN model using different compression ratios.

Measures / Compression	Feature number	Accuracy	F-measure	FP number	Testing time (s)	AANN training time (s)
0.9	52	0.89	0.85	149	0.00824	11.3572
0.8	46	0.89	0.85	166	0.00632	11.0828
0.7	40	0.90	0.86	156	0.00696	11.1471
0.6	35	0.90	0.86	158	0.00626	10.8626
0.5	29	0.89	0.85	165	0.00665	10.4815
0.4	23	0.88	0.84	162	0.00648	10.2219
0.3	18	0.88	0.84	200	0.00637	9.6991
0.2	12	0.87	0.82	191	0.00575	9.7039
0.1	6	0.82	0.76	324	0.00639	9.5528

Experiment 3. For EA-ANN model, I changed the value of *m* from 0 to 1.5 unevenly. One motivation is to explore how m influences the number of selected features and how the features size influences the classifier performance. Other motivations are same with the motivation described in Experiment 2 for AANN-ANN.

Results. The average results after 5-fold cross-validation are shown in Table 4. The testing time includes the time for features selection and time for prediction.

Table 4. The results of evaluation of the EA-ANN model using different *m*.

m	Feature number	Stop generation	Accuracy	F-measure	FP number	Testing time (s)	EA training time (s)
0	48	30	0.89	0.85	158	0.00698	32.4687
0.1	23	57	0.88	0.85	162	0.00620	56.6680
0.2	19	71	0.88	0.83	181	0.00561	71.4830
0.3	15	59	0.87	0.83	178	0.00599	50.3846

0.4	14	68	0.86	0.81	175	0.00579	69.4447
0.5	12	53	0.86	0.80	187	0.00559	50.4723
1	10	42	0.85	0.78	192	0.00575	37.2420
1.5	7	37	0.83	0.76	231	0.00524	35.6782

3.3 Discussion

Comparison among AANN-ANN, EA-ANN, ANN. The best AANN-ANN model and EA-ANN model are selected to compare with the ANN model. The numbers followed by AANN-ANN model is the compression ratio and compressed features size, while the numbers followed by EA-ANN is the *m* value and selected features size. The results are shown in Table 5.

Table 5. The results of evaluation of the ANN model, the best AANN-ANN model and the best EA-ANN.

	Accuracy	F-measure	FP number	Testing time
ANN	0.89	0.85	178	0.00738
AANN-ANN-0.7-40	0.90	0.86	156	0.00696
EA-ANN-0-48	0.89	0.85	158	0.00620

In terms of accuracy and F-measure, AANN-ANN improves the ANN model, because AANN extrudes some noises from the original dataset. However, it just improves 0.1 in each measure, we can conclude that the noises in dataset do not influence the performance much. In the other word, the noises do not deteriorate the result much though they are redundant. The best EA-ANN model is made with the *m*=0, that means we only evaluate individuals in EA by F-measure without any reduction pressure. That means some features can be removed because they are redundant, and F-measure will not be worse after removal. For FP number, we can see after feature selection and feature extraction, the FP number significantly decreases. We can deduce the features which indicate an email is a non-spam become more obvious after selection or extraction. However, the accuracy does not significantly increase, the FN number increases can be inferred. Thus, we can conclude the features which indicate an email is a spam email become not so obvious after feature selection or extraction. Combining the accuracy and F-measure results, we can think the process of feature extraction and selection used here is reconstructing the feature structure, strengthening some kinds of features and weakening some kinds of features. Therefore, the classifier performances better on some types of classification but worse on others and overall accuracy and F-measure do not improve much. The good thing is that what they strengthen is what we mostly care. So, they can be regarded as some improvements on basic ANN model. For testing time, both two modified models reduce the testing time, since the feature dimension is reduced, the number of connections in the ANN is decreased, so it needs less computation time than ANN with full features input. Further, AANN-ANN and EA-ANN can reduce the storage space for data. For example, initially, 57 features are needed to represent an instance. However, with AANN-ANN model, only about 40 features are needed, and the classification performance is better. Thus, we can just use 40 features to represent an instance, which reduces about 30% storage space. Similarly, the best EA-ANN model reduces about 16% storage space. Above two conditions are all based on the AANN-ANN and EA-ANN can give the same or better results with not using feature extraction or selection. If we pursue a large degree of reduction in storage space, we can use these two models with larger compress pressure and they both will give acceptable results. This will be discussed in next section.

Comparison between AANN-ANN and EA-ANN. The trends of accuracy and F-measure changing with selected features size of AANN-ANN and EA-ANN are plotted in the Fig. 6.

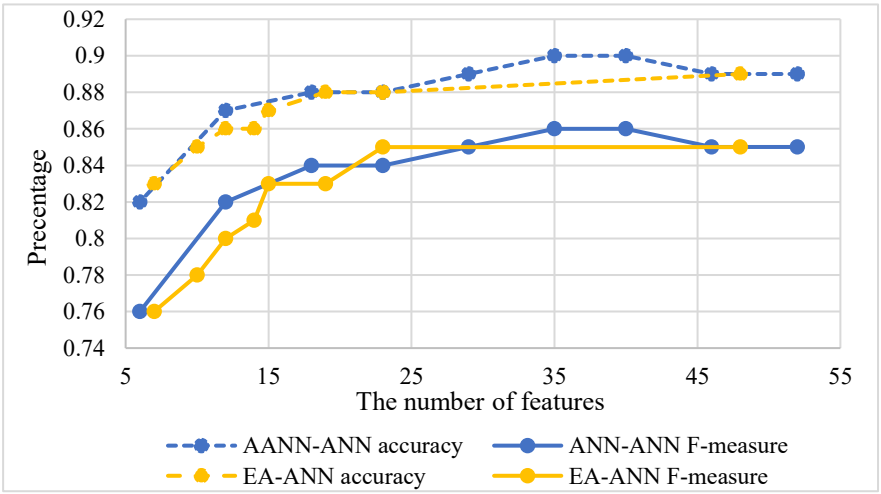

Fig. 6. The trends of accuracy and F-measure changing with selected features size of AANN-ANN and EA-ANN

For AANN-ANN model, we can directly find when the features size compressed to about 35-40 (60%-70% of original features), the model has the best performance. Performance does not change much during feature size is between 15 to 35 until I compressed feature size to about 15 features (30% of original features size), it starts to get worse significantly. For EA-ANN model, since the *m* cannot control the number of selected features precisely, I cannot get the evaluation of performance of model when features size is between 25-45. Beyond this, EA-ANN has the similar trend with AANN-ANN. We can deduce that if the compression ratio is too high, the noise feature would not be removed enough, the performance would not improve. If the compression ratio is too small, too many features are lost, classifier cannot get enough information to identify the instances, the performance will get worse. The reason for why performances have strong robustness (decrease slightly) before features reduced to 30% is that a large proportion of original features are 0. It is a very sparse and can be compressed largely. When compression ratio is under 30%, the AANN-ANN performs better than EA-ANN. Because AANN algorithm creates new features, it can combine several features into one feature. Though it has same features size with features selected by EA algorithm, it may contain more information than the latter. EA algorithm just selects original features, though some features contribute less than others, discarding them also causes information loss. For example, there are two features and their contributions are 0.8 and 0.2. AANN can fuse them using any linear or non-linear transformation, but EA just selects the feature with has 0.8 contribution.

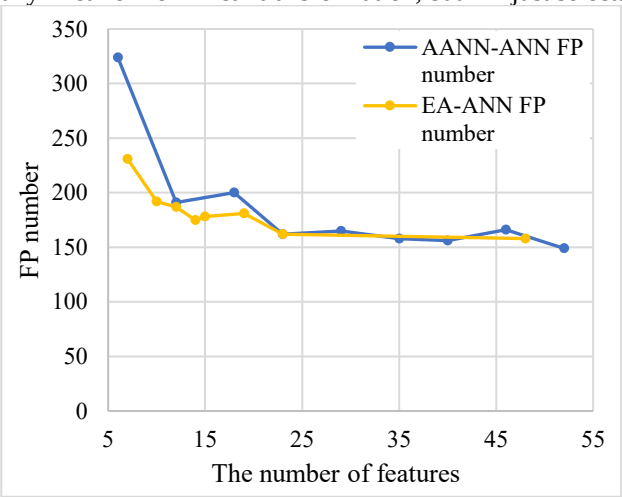

Fig. 7. The trends of FP number changing with selected features size of AANN-ANN and EA-ANN

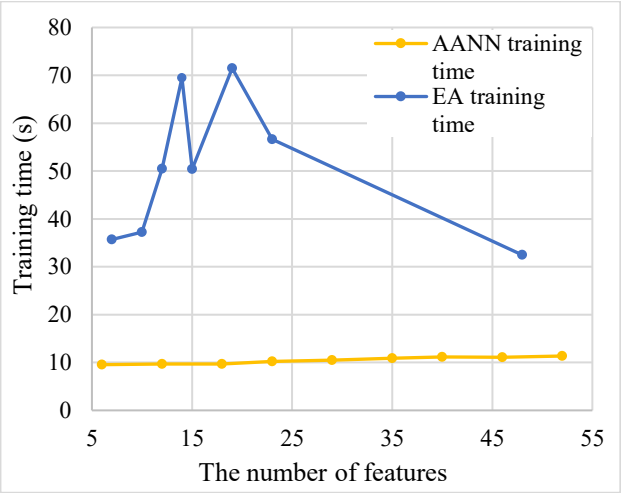

Fig. 8. The trends of training time changing with selected features size of AANN-ANN and EA-ANN

Regarding changing of FP numbers (Fig. 7), AANN-ANN still performs better than EA-ANN in a large degree of compression. The training time of AANN and EA is also worth to compare. From Fig. 8, we can observe that the training time of AANN rises steadily and slowly at about 10 seconds. On the contrary, the time consumed by EA is 3 to 7 times longer than AANN. Moreover, EA undergoes major changes and it seems no rules to follow. The explanation of this phenomenon is that EA is a randomised search, we cannot predict when it can find the good result and stop. Thus, EA seems much less reliable than AANN in this condition.

Comparison among ANN, AANN-ANN, EA-ANN and previous work. In paper [6], Hidalgo, Lopez and Sanz present an approach combines heuristics and a cost model to four machine learning algorithms: Naïve Bayes (NB), C4.5, PART and k-nearest neighbour (kNN). Three feature sets can be used are only words (W), only heuristic attributes (H) and both (WH). In spambase, words set mean the attributes 1-48, heuristic attributes are attributes 49-57. As for the four algorithms to be evaluated in this paper, the best performance of each algorithm is shown in Table 6. Since they are measured by accuracy, recall and precision, I calculated the F-measure for them in that they can be compared on F-measure with my models. I inserted my three models proposed in this paper into this table (highlighted by *) and listed all 7 approaches according to the accuracy ascending.

Table 6. The comparison of results of evaluation of algorithms in [6] and three models proposed in this paper.

	Accuracy	F-measure
NB-W	0.82	0.84
5NN-W	0.86	0.82
*ANN	0.89	0.85
*EA-ANN	0.89	0.85
*AANN-ANN	0.90	0.86
C4.5-WH	0.96	0.82
Part-WH	0.97	0.82

From this table, we can see three models proposed in this paper performs similarly. Though they do not have the highest accuracy, they are more balanced than other four algorithms, especially AANN-ANN model, because of higher

F-measure, about 0.85 to 0.86. Thus, my models are more robust for different dataset structure. The accuracies of my models are also acceptable, about 0.89 to 0.90, higher than NB-W and 5NN-W but less than C4.5-WH and Part-WH.

4 Conclusion and future work

This paper implemented an ANN model and designed two modified models, AANN-ANN and EA-ANN, based on it. They are used for the classification on spambase dataset. The noises in original spambase dataset have little influence on results, the best AANN-ANN model improves the accuracy of 0.1 and F-measure of 0.1 after extruding noises. The best performance of EA-ANN is same with the ANN. Though this is not like what I expect, the accuracy and F-measure improved by feature extraction or selection, the FP number is significantly reduced by two modified models. We can consider the processes of feature selection and extraction reconstruct the features structure and improved the performance of the task what we care (reduce the FP). Though the noises have little influence, but they are dispersive. In the other word, the original useful features are sparse. Until I compressed it to 0.3 of the original size, the performance starts to be worse significantly in both AANN-ANN and EA-ANN. With less than 0.3 of original data, AANN-ANN performs better than EA. Thus, we can conclude the quality of feature dimensionality reduction of AANN is better than EA. In terms of time, AANN-ANN and EA-ANN both shrink the testing time of ANN. The EA-ANN has the shortest testing time among three. However, the training time for EA is extremely unstable and 3 to 7 times longer than AANN. Overall, the best model is AANN-ANN with compression ratio being 0.7, which can achieve 0.90 accuracy and 0.86 F-measure with acceptable training time and testing time.

This work can be improved in many aspects. First, we can investigate what is the best batch size, which is what I did not explore in this paper. To implement it, we can select several key points of batch size and train and test model using values of key points, then do interpolation for the points between the key points, we can get a line fitting performance changed with batch size. Second, we can try different types of AANN, like bidirectional network, to compare the compression quality of different AANNs and figure out what is the most suitable AANN model for this problem. Third, for EA algorithm, different crossover and mutation methods can be attempted to enlarge search space to find the better solution.

References

[1] E. A. Erosheva and S. E. Fienberg, "Bayesian Mixed Membership Models for Soft Clustering and Classification," *Studies in Classification, Data Analysis, and Knowledge Organization Classification — the Ubiquitous Challenge*, pp. 11–26, 2005.

[2] K. Tretyakov, "Machine learning techniques in spam filtering", *Data Mining Problem-oriented Seminar*, MTAT. Vol. 3. No. 177. 2004.

[3] P. Cortez, C. Lopes, P. Sousa, M. Rocha, and M. Rio, "Symbiotic Data Mining for Personalized Spam Filtering," *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, 2009.

[4] Y. Yang and S. Elfayoumy, "Anti-Spam Filtering Using Neural Networks and Baysian Classifiers," *2007 International Symposium on Computational Intelligence in Robotics and Automation*, 2007.

[5] D. Dua and E. Karra Taniskidou, "Spambase", *UCI Machine Learning Repository: Spambase Data Set*, Available: https://archive.ics.uci.edu/ml/datasets/spambase. [Accessed: 29-Apr-2018].

[6] J. M. G. Hidalgo, M. M. López, and E. P. Sanz, "Combining text and heuristics for cost-sensitive spam filtering," *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning* -, 2000.

[7] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.

[8] M. L. Bermingham, R. Pong-Wong, A. Spiliopoulou, C. Hayward, I. Rudan, H. Campbell, A. F. Wright, J. F. Wilson, F. Agakov, P. Navarro, and C. S. Haley, "Application of high-dimensional feature selection: evaluation for genomic prediction in man," *Scientific Reports*, vol. 5, no. 1, 2015.

[9] G. James, *An introduction to statistical learning: with applications in R*. New York: Springer, 2014.

[10] M. Marseguerra and A. Zoia, "The autoassociative neural network in signal analysis: I. The data dimensionality reduction and its geometric interpretation," Annals of Nuclear Energy, vol. 32, no. 11, pp. 1191–1206, 2005.

[11] J. R. Dorronsoro, V. Lopez, C. S. Cruz and J. A. Siguenza, "Autoassociative neural networks and noise filtering," *IEEE Transactions on Signal Processing*, vol. 51, no. 5, pp. 1431-1438, May 2003. doi: 10.1109/TSP.2003.810276

[12] T. Marwala, "Fault classification in structures with incomplete measured data using autoassociative neural networks and genetic algorithm," *Current Science*, vol. 90, no. 4, pp. 542-8, Feb. 2006.

[13] T. D. Gedeon, J. A. Catalan, and J. Jin, "Image Compression using Shared Weights and Bidirectional Networks," *In Proceedings 2nd International ICSC Symposium on Soft Computing (SOCO'97)*, pp. 374-381.

[14] T. S. Li , " Feature selection for classification by using a GA-based neural network approach," *Journal of the Chinese Institute of Industrial Engineers*, vol. 23, no. 1, pp. 55–64, 2006.

[15] F. Gómez and A. Quesada, "Genetic algorithms for feature selection in Data Analytics | Neural Designer," *Neural Designer | Advanced analytics software*. [Online]. Available: https://www.neuraldesigner.com/blog/genetic_algorithms_for_feature_selection. [Accessed: 25-May-2018].