Automated Fish Classification Using Unprocessed Fatty Acid Chromatographic Data: A Machine Learning Approach

Jesse Wood[1], Bach Hoai Nguyen[1], Bing Xue[1], Mengjie Zhang[1], and Daniel Killeen[2]

[1] Victoria University of Wellington, Te Herenga Waka, Wellington, New Zealand
{jesse.wood, hoai.bach.nguyen, bing.xue, mengjie.zhang}@ecs.vuw.ac.nz
[2] New Zealand Institute for Plant and Food Research Limited, Nelson, New Zealand
daniel.killeen@plantandfood.co.nz

Abstract. Fish is approximately 40% edible fillet. The remaining 60% can be processed into low-value fertilizer or high-value pharmaceutical-grade omega-3 concentrates. High-value manufacturing options depend on the composition of the biomass, which varies with fish species, fish tissue and seasonally throughout the year. Fatty acid composition, measured by Gas Chromatography, is an important measure of marine biomass quality. This technique is accurate and precise, but processing and interpreting the results is time-consuming and requires domain-specific expertise. The paper investigates different classification and feature selection algorithms for their ability to automate the processing of Gas Chromatography data. Experiments found that SVM could classify compositionally diverse marine biomass based on raw chromatographic fatty acid data. The SVM model is interpretable through visualization which can highlight important features for classification. Experiments demonstrated that applying feature selection significantly reduced dimensionality and improved classification performance on high-dimensional low sample-size datasets. According to the reduction rate, feature selection could accelerate the classification system up to four times.

Keywords: AI applications · Classification · Feature selection · High-dimensional data · Particle Swarm Optimization · Multidisciplinary · Gas Chromatography · Fatty Acid

1 Introduction

Fish oil is rich in omega-3 polyunsaturated fatty acids, nutritionally important fats that are found at increasingly low concentrations in Western diets [22]. This has contributed to a high consumer demand for omega-3 supplements, produced from a wide range of marine biomass [17]. The suitability of a given fish species (or fish tissue) for the production of high-value omega-3 supplements depends on fatty acid composition, which is determined by an analytical chemistry technique called Gas Chromatography [6,20]. However, fatty acid data must be

carefully processed and interpreted by domain experts (i.e. chemists), which is very expensive and time-consuming. Previous works using CNNs, [3,14], showed high classification accuracy on Gas Chromatography data. However, these black-box models do not produce interpretable models, making it difficult to verify/troubleshoot these models for fish processing in a factory setting.

The goal of this work is to automate the processing and interpretation of Gas Chromatography data using machine learning algorithms, to substantially increase fatty acid analysis throughput. However, it is not a trivial task to format Gas Chromatography data for existing classification algorithms. Furthermore, each Gas Chromatography data consists of almost 5000 values (features/variables), far more numerous than the number of fish samples (153). This large number of features relative to samples (the curse of dimensionality) results in a sparsely populated data space, which can result in overfitting i.e. where the built model works well on the training set but poorly on the test (unseen) set. Redundant (providing the same information as other features) or irrelevant features (providing misleading information for the classification task) are also common in this type of dataset [15], which can reduce classification performance and cause long training times. Therefore, the paper also assessed the utility of feature selection to preprocess and remove these irrelevant/redundant features.

The goals of this work are to investigate the viability of classifying different marine biomass, automate processing of raw Gas Chromatography data, improve analytical throughput and reduce labour costs, and reduce the dimensionality of Gas Chromatography data required to perform fish oil production and analysis. The contributions of this work are broken into three main steps:

- Data preprocessing: This step converts Gas Chromatography data into tabular format data appropriate as input into a machine learning algorithm. The paper finds an effective method to detect and fill the missing packets/features which improves the classification performance over using the raw data.
- Analysing classification algorithms: The second step performs experiments with five types of classification algorithms, including instance-based classifiers, probabilistic classifiers, tree-based classifiers, ensemble classifiers, and kernel-based classifiers, to classify fish samples [4,7,8,9,13]. Experiments find that kernel-based classifiers, particularly linear SVM, achieve high classification accuracy on the fish data. The paper visualises the learnt model and identifies that not all the data, represented as *features*, are useful, which leads to the final step.
- Feature selection: The last step applied feature selection methods to reduce the amount of collected data i.e., the number of features. The experimental results illustrate that the number of features could be reduced by almost 75% while improving the classification performance.

2 Gas Chromatography

Gas Chromatography is an analytical chemistry method commonly used to investigate the fatty acid compositions of biological samples e.g. marine oils [6,20].

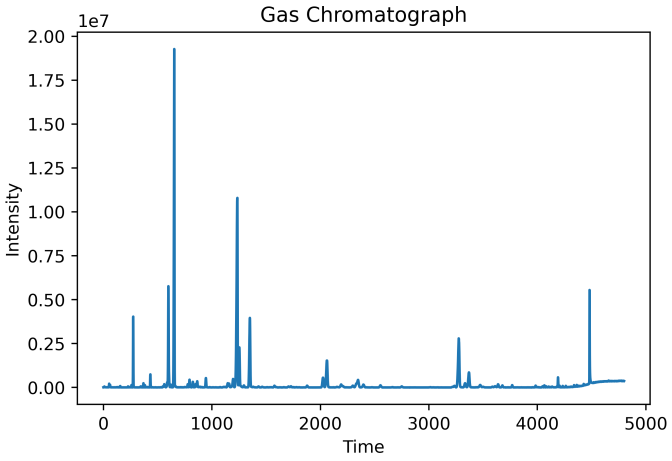


Fig. 1: Gas Chromatogram of Fatty Acid Methyl Esters from Snapper Skin.

It works by increasing the temperature of a very narrow 'capillary' column, which separates each fatty acid from the complex mixture based on their individual chemical characteristics e.g. molecular size, volatility, and polarity. An example of Gas Chromatography for fatty acid analysis is shown in Figure 1. The x-axis represents the time required to separate the individual fatty acids (or a packet), and the y-axis represents peak intensity (or the packet intensity), which is proportional to the concentration of each fatty acid. Chemists integrate the area under each peak to measure how much of each fatty acid is present, and use this information to understand the best use of the oil. This process can be slow, labour-intensive and expensive.

The goal of this work is to apply machine learning, particularly classification algorithms to automatically classify the fish data, a real-world problem in New Zealand. However, the current Gas Chromatography data is not readily applied to machine learning algorithms due to missing packets which are not caught by the system detector. The missing packets cause the misalignment between two samples, i.e., intensities at the same time of the two chromatographs may have different meanings. Therefore, it is necessary to detect such missing packets to align the data before applying machine learning algorithms.

3 Data Preprocessing and Formation of Classification Problems

The Y-data output from the Gas Chromatography analysis consists of many packets with variable intensities. In theory, they could be used as features to classify the different fish samples, but there were a large number of inconsistencies between packets in the different fish samples. An example, focusing on these

Table 1: Inconsistent Timestamps

	Timestamp		
	Sample 1	Sample 2	Sample 3
Packet 1	51	50	50
Packet 2	52	51	51
Packet 3	53.05	53.1	53

inconsistencies for three different fish samples, is shown in Table 1. Although all three samples have three packets, their timestamps are different. For example, the timestamp of the first packet of Sample 1 is 51, while the timestamp of the first packet of Sample 2 is 50. In other words, the first packet of Sample 1 does not correspond to the first packet of Sample 2, and thus it does not make sense to directly apply a classification algorithm to the raw data. Initial experiments tried KNN (K=3), and the classification performance was only 67%, which is quite low.

Further investigation revealed that the main reason was due to the missing packets, caused by the absence of signal at the Gas Chromatography detector. For example, for Sample 1, the packet at the timestamp 50 is missed, and thus the first packet of Sample 1 is at 51. These missing packets are unavoidable for this dataset, therefore a method is needed to handle missing data. Preprocessing aligns the packets from all the samples. Firstly, all unique timestamps are collected by analysing all the possible samples in the training set. For the example given in Table 1, the set of unique timestamps is {50, 51, 52, 53, 53.05, 53.1}. Thus, there should be six packets in total, while Table 1 shows only three packets for each sample. Based on the timestamp set, the packets at {50, 53, 53.1} are missing for Sample 1. Once the missing packets are identified, these missing intensities need to be filled.

This work tried three different standard methods for missing values: filling 0, filling the average value, and filling the median value. The results show that filling 0 gives the most promising results with 83.57% on KNN (K=3). The possible reason is that the missing packets have low intensities, which the detector might not be able to detect. Thus, the 0 value is quite close to the intensities of the missing packets. Therefore, the filling 0 method was chosen. The authors are aware that there are more complex methods for imputing missing values, [23,25], but they are not the focus of the paper and will be left for future work [24].

The processing gives 4800 packets for each sample, which meant each sample had *4800* features. The number of fish samples was 153. There is a class imbalance for the fish species dataset, where Blue cod is the majority class e.g., 68 samples are Blue Cod of the total 153 samples. There are two classification tasks associated with the data:

- To predict the fish species for each fish sample. There are four fish species: *Snapper, Gurnard, Tarakihi*, and *Blue cod*.
- To predict from which body part the fish sample is extracted. There are six body parts: *Frame, Gonad, Head, Liver, Skin*, and *Guts*.

4 Classification Performance

The following section illustrates the classification performance on the fish species and body parts.

4.1 Experiment Settings

Firstly, since the number of samples is small, the experiment uses 10-fold cross-validation to conduct the experiments. For 10-fold cross-validation, the method divides the data into 10 folds such that the proportions of the classes in each fold are representative of the proportions in the whole dataset. Each fold plays the testing role, while the remaining 9 folds are combined to form a training set. A classification algorithm is then trained on the training set, and the obtained classifier is evaluated on the test set. Finally, 10 testing accuracies are obtained, and their mean value and standard deviation are given as the final classification performance. The experiment measures the balanced accuracy, so as not to bias results towards the majority class (i.e. Blue cod for fish species).

These experiments compare five well-known classifications: K Nearest Neighbours (KNN), Naive Bayes (NB), Random Forest (RF), Decision Trees (DT), and Linear Support Vector Machines (SVM) [7,8,9,13,4]. The parameters are the default settings in *scikit-learn* [19].

4.2 Results and Discussion

Table 2: Classification Accuracies

Dataset	Method	AvgTrain ± Std	AveTest ± Std
Fish Species	KNN	83.57 ± 1.80	74.88 ± 12.54
	RF	100.0 ± 0.00	85.65 ± 10.76
	DT	100.0 ± 0.00	76.98 ± 13.12
	NB	79.54 ± 1.60	75.27 ± 4.35
	SVM	**100.0 ± 0.00**	**98.33 ± 5.00**
Body Parts	KNN	68.95 ± 3.49	43.61 ± 13.48
	RF	100.00 ± 0.00	72.60 ± 16.15
	DT	100.00 ± 0.00	60.14 ± 14.57
	NB	65.54 ± 2.69	48.61 ± 12.19
	SVM	**100.00 ± 0.00**	**79.86 ± 8.52**

Table 2 shows the results for KNN, RF, DT, NB, and SVM. Results are given for fish species (top), and fish part (bottom) datasets. The mean and standard deviation of balanced accuracy is given using the fish species and part datasets. For each dataset, the best accuracy is emphasized in bold.

As can be seen from the table, RF, DT and SVM achieve 100% training accuracies. However, on the test set, DT and RF do not achieve good classification performance. The main reason is that there is a small number of training

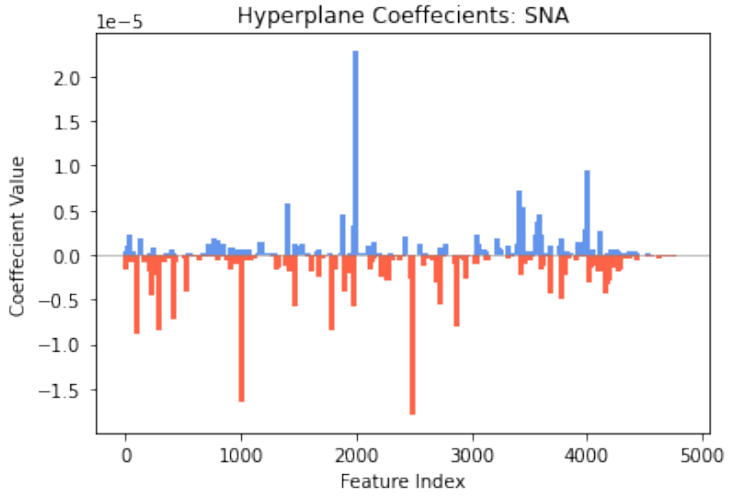
samples. The trees built by DT and RF can perfectly fit the training data by creating large trees that remember all the possible training samples. Such trees do not generalise well on the test set, which is the overfitting problem in machine learning. KNN does not achieve good performance since it is a distance-based classification algorithm which suffers the most from the large number of features. Similar to KNN, NB does not achieve good performance since it assumes conditional independence between features that may not be true in the fish datasets. The SVM classifier outperforms the other classifiers on the test set, with 98.33% and 79.86% for fish species and body parts, respectively. The main reason is that SVM can handle a large number of features, so SVM is suitable to classify the fish data.

Another essential point is that the classification accuracy on the fish species is always higher than the classification accuracy on the body parts. The results suggest that classifying body parts is a more challenging problem. A possible reason is that the tissue samples from different species may have very different chemical components. Meanwhile, the tissue samples from different body parts (but on the same fish species) may have similar chemical components. Future work will investigate more sophisticated mechanisms to improve the classification performance on classifying body parts.

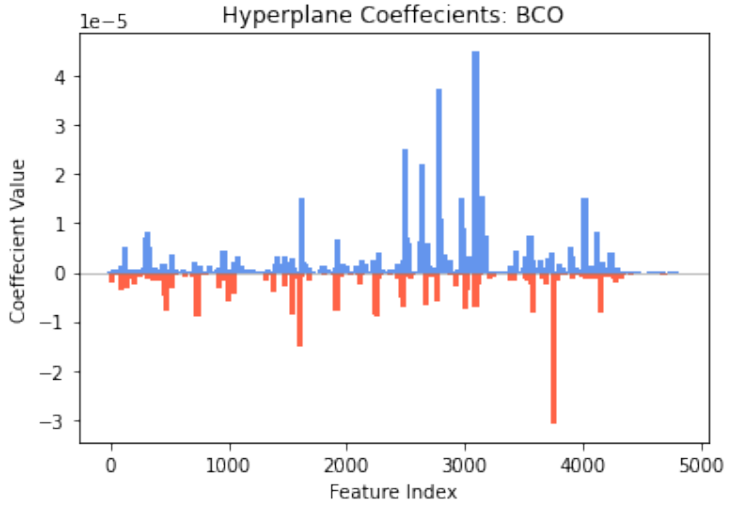
4.3 Interpret SVM models

Achieving a high classification performance is great. However, in real-world applications, it is essential to analyse why the models work well. This subsection analyzes the Linear SVM model built to classify the fish species. The main idea of SVM is to build hyperplanes that separate different fish species. For SVM with linear kernels, the hyperplane is represented by a weight vector in which each weight is associated with a feature. The larger the weight, the more important the corresponding feature. After an SVM classification algorithm is trained on the training set, an SVM classifier containing a learned weight vector is obtained. This section analyses the learned weight vector to examine the contribution of each packet/feature.

Figures 2a and 2b show the coefficients of hyperplanes to separate Snapper and Blue cod from other species, respectively. The horizontal axis is the feature index and the vertical axis is the coefficient value. The negative weights are in red and the positive weights are in blue. Gas Chromatography data is non-negative, so only negative weights push toward the negative class, therefore positive weights are expected values, and the negative values are not. Note that when considering the feature importance, the absolute values of the weights should be considered, i.e., the longer the bar, the more important the corresponding features. Both figures demonstrate that most features have relatively small weights, which suggests not all the 4800 packets/features are needed to classify the fish data.



(a) Snapper



(b) Blue cod

Fig. 2: SVM Hyperplane Coefficients

5 Feature Selection

5.1 Motivations

As can be seen in the SVM models, it is not necessary to use all the 4800 packets/features to perform fish classification. Therefore, the number of packets can be reduced while maintaining (or even improving) the classification performance. In an automated classification system, it would be great to significantly reduce the number of packets. Since then the system will not need to wait for a large number of packets to arrive at the end of the detector, hence significantly improving the system efficiency and throughput. The remaining question is which packets or features should be used. This question motivates us to conduct a further investigation using feature selection to select the most important packets/features.

5.2 Feature Selection Methods

In a classification problem, the classification performance relies heavily on feature quality. However, in a large set of features as in the fish data, there are usually redundant or irrelevant features that blur useful information provided by the relevant features. Feature selection aims to select an informative subset of relevant features, which is expected to significantly reduce the number of features while maintaining (or even improving) the classification performance. In a feature selection system, subset evaluation is an essential component that evaluates the quality of a feature subset. Based on the subset evaluation, the system can continuously improve the subset quality until a stopping criterion is met. The final feature subset is the output as the final solution.

This section compares four common feature selection methods:

- $\boldsymbol{\chi^2}$ (chi-square) [12] is a statistical measure that computes the independence of two variables X and Y. The formula of χ^2 is

$$\chi^2 = \sum_{k=1}^{N} \frac{(X_k - Y_k)^2}{Y_k} \tag{1}$$

  where k is the index of the sample and N is the number of samples. In feature selection, χ^2 can be used to measure the independence between a feature and a class label. Since there is usually a high dependency between a relevant feature and a class label, the low χ^2 value indicates that the features are more relevant. Thus, the features can be ranked in ascending order and the top-ranked features can be selected.
- **Minimum Redundancy and Maximum Relevance (mRMR)** [5] uses mutual information to perform feature selection. Mutual information between two variables X and Y, i.e., $I(X;Y)$ calculates the dependency between two or more variables. mRMR aims to select a feature subset such that the redundancy of the selected features is minimised and the relevance

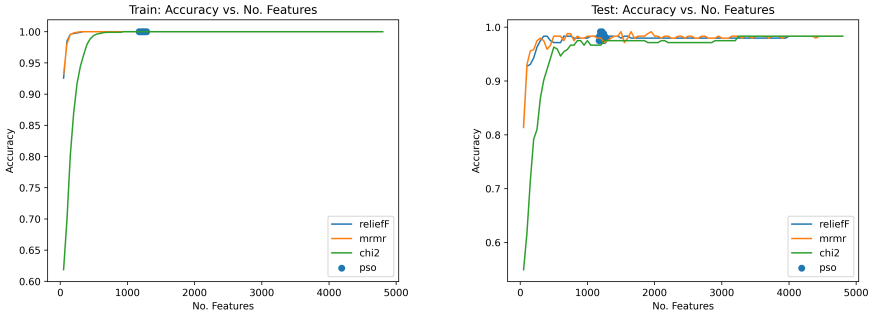
between the selected features and the class label is maximised. Given a set of selected features A, the score of a feature X_i, i.e., S_i is calculated by the following formula:

$$S_i = I(Y; X_i) - \frac{1}{|S|} \sum_{X_j \in A} I(X_i; X_j) \quad (2)$$

mRMR has many iterations where at each step mRMR will add the best feature based on Eq. (2). mRMR stops when a predefined number of features are selected.

- **ReliefF** [21] is a feature selection algorithm based on distance measures. In ReliefF, a good feature should be able to separate instances from different classes well while the instances from the same class should not be far from each other. The algorithm ranks all features based on the idea of nearest neighbours. For a feature, if the distance between two nearest instances from *different* classes (a miss) is large, the feature score is increased since the feature can separate different classes well. On the other hand, if the distance between the two nearest instances from the *same* class is large (a hit), the feature score is decreased. In ReliefF, the higher the score, the more relevant the feature. Therefore, all features are ranked in descending order, and the top-ranked features are selected.
- **Particle Swarm Optimisation (PSO) [10,16] for Wrapper Feature Selection** utilises the classification performance as the fitness function to achieve feature selection. The main idea is to have a swarm of particles that can explore the feature subset space in parallel. Each particle represents a feature subset. The quality of each particle is the classification performance of the corresponding feature subset. Since it is necessary to train a classification algorithm during the evaluation process, the classification algorithm is "wrapped" inside the PSO algorithm (that is why the algorithm is called Wrapper PSO). In this work, a linear SVM is used as the wrapped classification algorithm since it achieves good classification performance. Each particle records the best feature subset that it discovered so far (called personal best or *pbest*) and the best feature subset that is discovered by the whole swarm so far (called global best or *gbest*). The particle then updates its position by moving towards the two best positions. It is expected that the new subset at the new position will have better quality (i.e., higher classification performance) than the previous position. An advantage of PSO is that the particle movement is stochastic. Thus, the swarm can globally explore the feature subset search space, which is an essential point when dealing with a large and complex search space like feature selection. Therefore, PSO has gained much attention from the feature selection community recently [15].

Although there are other advanced and complicated feature selection algorithms [11,26,1,2], this work starts with the above four simple but well-known techniques. If the results are promising, future work will investigate extensions of these and/or other feature selection algorithms.



(a) Species: Training set (b) Species: Test set

Fig. 3: Classification Accuracy of Fish Species on Different Numbers of Selected Features.

5.3 Experiment Settings

Following the same setting in the classification part, this experiment uses 10-fold cross-validation to generate the training and test sets. For each method, the balanced classification accuracy is measured with a linear SVM classification algorithm [18]. For χ^2, mRMR, and ReliefF, a hyperparameter for the number of selected features must be given. Therefore, the experiments measure the performance of the three algorithms on a wide range of the number of features: {50, 100, 150, ..., 4800} with increment 50. For PSO, the swarm size is set to 30 and the maximum number of iterations to 100. An advantage of PSO is that it does not need to specify a hyperparameter for the number of selected features. Since PSO is a stochastic algorithm, it is run 30 independent times on each classification task to make a reliable comparison.

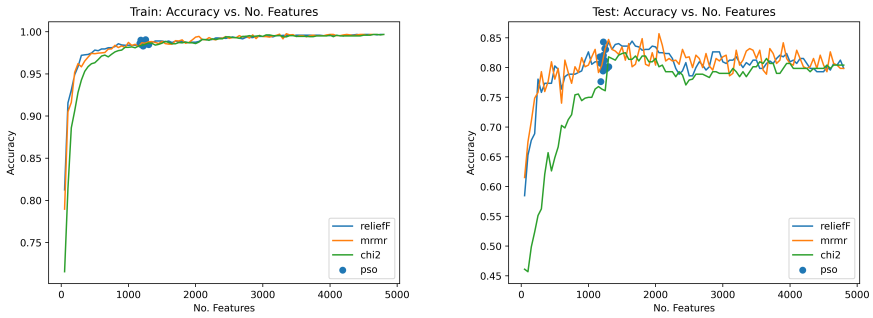
5.4 Feature Selection Performance on Fish Species Classification

Figure 3 shows the results for χ^2 (chi2), ReliefF, mRMR and PSO on the fish species. The vertical axis is the classification accuracy and the horizontal axis is the number of selected features. As can be seen from the figures, the three algorithms χ^2, mRMR, and ReliefF perform poorly when the number of selected features is small. The main reason is that when the number of selected features is small, many relevant features are not selected, and thus essential classification information is missed. Among the three algorithms, χ^2 usually achieves the lowest classification performance since χ^2 does not reduce the feature redundancy and does not consider the interactions between features. ReliefF and mRMR achieve comparative performance. mRMR achieves its highest training and testing accuracies when the number of selected features is around 1500, which can be seen in Table 3.

As can be seen from the figure, most feature subsets evolved by PSO have from 1100 to 1500 features. The results indicate that PSO can automatically

Table 3: Best accuracy on Fish Species.

Method	Number of features	Training accuracy	Testing accuracy
ReliefF	359	100.0	98.33
mRMR	**1500**	**100.0**	**99.17**
χ^2	3250	100.0	98.33
PSO	**1192**	**100.0**	**99.17**
Full	4800	100.0	98.33



(a) Part: Training set (b) Part: Test set

Fig. 4: Classification accuracy of Fish Body Parts on Different Numbers of Selected Features.

determine a good number of selected features, which cannot be achieved by the other three algorithms. As can be seen in Table 3, the highest classification performance of PSO is 99.17% which is about 1% higher than using all features. Meanwhile, PSO can remove 75% of the features, which means the classification system can be four times faster given the number of required packets/features is reduced by four times.

5.5 Feature Selection Performance on Body Parts Classification

Figure 4 shows the results for χ^2, ReliefF, mRMR and PSO on the fish part dataset. As can be seen in Figure 4a, χ^2, mRMR, and Relief-F witness a sharp improvement when the number of selected features is in the range [0, 500], which indicates that the 500 top-ranked features are essential to select. After that, the three approaches have a gradual incline, which peaks at 100% where all the features are selected. On the other hand, PSO selected feature subsets with sizes ranging in [1200,1300]. Given the same classification performance, PSO usually selects a smaller number of features than the other three feature selection algorithms. The main reason is that PSO considers the interaction in the whole set of features, meanwhile, the other algorithms only consider the pair-wise interactions between feature pairs.

Table 4: Best Accuracy on Fish Body Parts

Method	Number of features	Training Accuracy	Testing Accuracy
ReliefF	1650	100.0	84.44
mRMR	**1500**	**100.0**	**86.94**
χ^2	1550	100.0	82.50
PSO	1223	100.0	84.31
Full	4800	100.0	79.86

Table 4 illustrates the best accuracy for classifying fish body parts. As can be seen from the table, the best classification performance at 86.94% is achieved with 1500 features selected by mRMR. Thus, feature selection can also improve 7% accuracy over using all features. Meanwhile, the number of features is reduced by 2.5 times, which means the system can be 2.5 times faster. It should be noted that the testing performance of PSO is not as good as mRMR despite its superior training performance. The results indicate the potential overfitting of PSO on classifying body parts, which can be investigated more in the future.

5.6 Summary

In general, feature selection can significantly reduce the number of required packets/features and improve classification performance. For classifying the fish species, 75% of packets can be removed. For classifying the body parts, 60% of packets can be removed. The significant reduction means that the overall classification system can be up to 4 times faster. It should be noted that classifying the body part is more challenging than classifying the fish species. That is why classifying the body parts requires more features. Last but not least, PSO can automatically determine a good number of selected features. In general, PSO achieves good classification performance, except for some signs of overfitting which can be investigated in future.

6 Conclusions and Future Work

This paper has proposed an interpretable and effective classification process for fish oil analysis. Based on the results, it can be concluded that machine learning is a promising direction to improve the effectiveness and efficiency of the overall fish product system. In terms of accuracy, the proposed model can achieve high classification performance on classifying both fish species and body parts. However, fish species are easier to predict than body parts since there is more intra-class variation within fish species than there is a similarity between the same part from different fish. Among the considered classification algorithms, linear SVM achieves the best classification performance since it is suited to high-dimensional problems. Analysis of the SVM model demonstrates that not all packets are needed, and thus feature selection has been conducted to significantly reduce the number of packets and improve the classification performance.

It is worth noting that the classification and feature selection methods presented in this paper could be extended to further improve performance. This is particularly useful for the lower-accuracy fish part dataset. A potential direction is to improve the classification performance by constructing more informative high-level features, also known as feature construction. In addition, a more sophisticated imputation method can be developed to fill the missing packets in the fish data.

References

1. Alsahaf, A., Petkov, N., Shenoy, V., Azzopardi, G.: A framework for feature selection through boosting. Expert Systems with Applications **187**, 115895 (2022)
2. Alweshah, M., Alkhalaileh, S., Al-Betar, M.A., Bakar, A.A.: Coronavirus herd immunity optimizer with greedy crossover for feature selection in medical diagnosis. Knowledge-Based Systems **235**, 107629 (2022)
3. Bi, K., Zhang, D., Qiu, T., Huang, Y.: Gc-ms fingerprints profiling using machine learning models for food flavor prediction. Processes **8**(1), 23 (2020)
4. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3), 273–297 (1995)
5. Ding, C., Peng, H.: Minimum redundancy feature selection from microarray gene expression data. Journal of bioinformatics and computational biology **3**(02), 185–205 (2005)
6. Eder, K.: Gas chromatographic analysis of fatty acid methyl esters. Journal of Chromatography B: Biomedical Sciences and Applications **671**(1-2), 113–131 (1995)
7. Fix, E., Hodges, J.L.: Discriminatory analysis. nonparametric discrimination: Consistency properties. International Statistical Review/Revue Internationale de Statistique **57**(3), 238–247 (1989)
8. Hand, D.J., Yu, K.: Idiot's bayes—not so stupid after all? International statistical review **69**(3), 385–398 (2001)
9. Ho, T.K.: Random decision forests. In: Proceedings of 3rd international conference on document analysis and recognition. vol. 1, pp. 278–282. IEEE (1995)
10. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks. vol. 4, pp. 1942–1948. IEEE (1995)
11. Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J., Liu, H.: Feature selection: A data perspective. ACM computing surveys (CSUR) **50**(6), 1–45 (2017)
12. Liu, H., Setiono, R.: Chi2: Feature selection and discretization of numeric attributes. In: Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence. pp. 388–391. IEEE (1995)
13. Loh, W.Y.: Classification and regression trees. Wiley interdisciplinary reviews: data mining and knowledge discovery **1**(1), 14–23 (2011)
14. Matyushin, D.D., Buryak, A.K.: Gas chromatographic retention index prediction using multimodal machine learning. Ieee Access **8**, 223140–223155 (2020)
15. Nguyen, B.H., Xue, B., Zhang, M.: A survey on swarm intelligence approaches to feature selection in data mining. Swarm and Evolutionary Computation **54**, 100663 (2020)

16. Nguyen, H.B., Xue, B., Andreae, P., Zhang, M.: Particle swarm optimisation with genetic operators for feature selection. In: 2017 IEEE Congress on Evolutionary Computation (CEC). pp. 286–293 (2017). https://doi.org/10.1109/CEC.2017.7969325
17. Panse, M.L., Phalke, S.D.: World market of omega-3 fatty acids. Omega-3 Fatty Acids pp. 79–88 (2016)
18. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
19. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. the Journal of machine Learning research **12**, 2825–2830 (2011)
20. Restek: High-resolution gc analyses of fatty acid methyl esters (fames)
21. Robnik-Šikonja, M., Kononenko, I.: Theoretical and empirical analysis of relieff and rrelieff. Machine learning **53**(1), 23–69 (2003)
22. Simopoulos, A.P.: Evolutionary aspects of diet: the omega-6/omega-3 ratio and the brain. Molecular neurobiology **44**(2), 203–215 (2011)
23. Tomasi, G., Van Den Berg, F., Andersson, C.: Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data. Journal of Chemometrics: A Journal of the Chemometrics Society **18**(5), 231–241 (2004)
24. Tran, C.T., Zhang, M., Andreae, P.: Multiple imputation for missing data using genetic programming. In: The Annual Conference on Genetic and Evolutionary Computation. pp. 583–590 (2015)
25. Zhang, D., Huang, X., Regnier, F.E., Zhang, M.: Two-dimensional correlation optimized warping algorithm for aligning gc× gc- ms data. Analytical Chemistry **80**(8), 2664–2671 (2008)
26. Zhang, Y., Gong, D.w., Gao, X.z., Tian, T., Sun, X.y.: Binary differential evolution with self-learning for multi-objective feature selection. Information Sciences **507**, 67–85 (2020)