Instance-recognition for detecting batches of marine biomass using rapid evaporative ionisation mass spectrometry

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. Instance recognition in marine biomass processing presents a unique challenge in quality control and traceability. This study explores a comprehensive approach to detecting whether two instances of marine biomass were processed in the same batch, employing a diverse array of machine learning techniques. We compare traditional machine learning methods, evolutionary computation via multi-tree Genetic Programming with multiple class-independent feature construction, and advanced deep learning models including Transformers, Convolutional Neural Networks (CNNs), Long-short Term Memory (LSTMs), Variational Autoencoders (VAEs), Mamba, and Kolmogorov-Arnold Networks (KANs). Our methodology encompasses both traditional binary classification and contrastive learning approaches, with a particular focus on Siamese networks for pairwise comparisons. Results demonstrate that contrastive learning, especially when implemented through Siamese networks, consistently outperforms traditional binary classification across various model architectures. This research not only advances the field of marine biomass instance recognition but also provides insights into the comparative efficacy of diverse machine learning paradigms in tackling complex recognition tasks.

Keywords: AI applications · contrastive learning · explainable AI · machine learning · marine biomass · mass spectrometry · multidisciplinary AI

1 Introduction

The ability to accurately identify and trace the processing history of marine biomass is crucial for ensuring quality control, maintaining regulatory compliance, and optimizing production processes in the marine biotechnology industry. Instance recognition, particularly determining whether two samples were processed in the same batch, presents a unique challenge due to the subtle variations in biomass characteristics and processing conditions. Traditional methods

of batch identification often rely on manual record-keeping or simplistic physical or chemical markers, which can be prone to errors or manipulation. In recent years, the advent of sophisticated machine learning techniques has opened new avenues for addressing complex recognition tasks across various domains. However, applying these advanced methods to marine biomass instance recognition remains relatively unexplored. This research aims to bridge this gap by investigating the efficacy of a wide spectrum of machine learning approaches in solving the instance recognition problem for marine biomass batches. Our study employs a multi-faceted approach, leveraging:

1. **Traditional machine learning algorithms**, providing a baseline for comparison. Specifically, Random Forest (RF) [16], K-Nearest Neighbors (KNN) [12], Decision Trees (DT) [7], Naive Bayes (NB) [14], Logistic Regression (LR) [20], Support Vector Machines (SVM) [10], and Linear Discriminant Analysis (LDA) [2], OPLS-DA [6].
2. **Ensemble method** [15]: A combination of the above traditional methods.
3. **Evolutionary computation techniques**, specifically multi-tree Genetic Programming [22] with multiple class-independent feature construction . [29, 30], which offers the potential for discovering novel, interpretable features.
4. **State-of-the-art deep learning models**, including Transformers [11, 31], (CNNs) [23–26]; Long-short Term Memory (LSTMs) [17]; Variational Autoencoders (VAEs) [19] ; Mamba [13]; and Kolmogorov-Arnold Networks (KANs) [27].

A key innovation in our approach is the application of contrastive learning techniques, particularly through Siamese networks, to the task of instance recognition. By framing the problem as a pairwise comparison rather than a straightforward binary classification, we hypothesize that models can learn more nuanced and robust representations of batch similarities and differences.

This research not only contributes to the specific field of marine biomass processing but also offers broader insights into the comparative strengths of various machine learning paradigms when applied to complex recognition tasks. By systematically evaluating these diverse approaches, we aim to identify the most effective strategies for instance recognition, potentially informing future applications in related domains.

In the following sections, we detail our methodological approach, present our experimental results, and discuss the implications of our findings for both the marine biotechnology industry and the wider field of machine learning-based instance recognition. Our comparative analysis of traditional, evolutionary, and deep learning methods, including the novel application of KANs in this domain, provides a comprehensive evaluation of current capabilities in tackling the challenges of marine biomass instance recognition.

2 Related Works

Having established the importance of instance recognition in marine biomass processing and outlined our multi-faceted approach, we now turn to an examination of related works that inform and contextualize our research.

2.1 Rapid Evaporative Ionisation Mass Spectrometry

Rapid evaporative ionisation mass spectrometry (REIMS) [3] has shown promise in beef processing, where it was able to detect horse meat contamination in beef. Most impressively, horse meat contamination was detected at 1-5% - very low levels [4]. This demonstrates the REIMS technique is incredibly sensitive to contamination. REIMS has been applied to fish fraud detection to identify fish species and identify catch methods for fish products [5]. The method was so accurate it was able to identify incorrectly labelled instances in the training data. The analysis of biomass in the literature - [3–5] - has shown that Orthogonal Partial Least Squares Regression Discriminant Analysis (OPLS-DA) [2, 6, 9] for binary classification combined with Principal Component Analysis [1] for dimensionality reduction is an effective technique for REMIS analysis. These works rely on outlier thresholding with manually set thresholds for outlier detection, which are chosen by experts with domain expertise in the application field.

The deep learning methods explored in this work outperform the OPLS-DA method that is traditionally used in the literature. This work introduces instance recognition for batch detection, a novel application of REIMS that has not been explored in the field before. This work looks to expand on the existing literature and explore a variety of deep learning and evolutionary computation techniques, as well as traditional machine learning techniques. Instead of using outlier thresholding with manually set hyperparameters for thresholds, this work uses contrastive learning with learnable feature representation accompanied by learnable thresholding parameters.

2.2 Siamese Networks and Contrastive Learning

Siamese neural networks, proposed in 1993 by LeCun [8], and prominent today in deep learning for object detection and segmentation [18], and ransomware classification [32]. Siamese neural networks are a type of contrastive learning. Contrastive learning is unsupervised or supervised learning where the goal is to learn a similarity metric between two inputs, by contrasting them with other inputs. A similar method concept to the thresholding method for detecting outliers was previously mentioned in [4, 5].

This work provides a modern approach to Siamese networks for supervised contrastive learning, updating the framework to utilize modern state-of-the-art deep learning methods and evolutionary computation. We explore Siamese networks for Convolutional Neural Networks (CNNs), Transformers, Long-Short Term Memory (LSTM) Variational Autoencoders (VAE), Kolmogorov-Arnold Networks (KANs) and Mamba. We evaluate the contrastive learning approach

compared to the vanilla binary classification task with those same networks. The experimental results show significant improvements for instance recognition of batches of fish with Siamese networks and contrastive learning when compared to vanilla binary classification.

2.3 Deep Learning Methods

The field of deep learning has seen the development of various neural network architectures, each designed to address specific challenges in data processing and representation learning. Transformers [11, 31] have revolutionized natural language processing with their ability to capture long-range dependencies in sequential data, enabling breakthroughs in tasks such as machine translation and text generation. Convolutional Neural Networks (CNNs) [23–26], originally inspired by the visual cortex, excel at spatial feature extraction and have become the backbone of many computer vision applications. For handling time-series data and learning long-term dependencies, Long Short-Term Memory (LSTM) networks [17] have proven particularly effective, finding applications in speech recognition and sentiment analysis. Variational Autoencoders (VAEs) [19] have emerged as powerful tools for learning robust latent representations of data, facilitating tasks such as image generation and anomaly detection. More recently, the Mamba architecture [13] has introduced innovations in sequence modelling, promising improved efficiency and performance over traditional recurrent models. Complementing these approaches, Kolmogorov-Arnold Networks (KANs) [27] leverage the universal approximation theorem to model complex functions, offering a theoretically grounded approach to neural network design. Each of these architectures brings unique strengths to the table, and their combined advancements have significantly expanded the capabilities and applications of deep learning across various domains.

3 Method

Building upon these foundational studies and leveraging the latest advancements in machine learning, we now present our comprehensive methodological approach to the challenge of marine biomass instance recognition.

3.1 Binary Classification

Binary classification is a fundamental task in machine learning and statistical analysis, where the goal is to categorize input data into one of two mutually exclusive classes or categories. Formally, given an input vector $\mathbf{x} \in \mathbb{R}^n$, a binary classifier aims to learn a function $f : \mathbb{R}^n \to \{0, 1\}$ that maps $\mathbf{x}$ to a binary label $y \in \{0, 1\}$. This problem arises in numerous real-world applications, including spam detection, medical diagnosis, and sentiment analysis. Common approaches to binary classification include traditional machine learning methods such as Random Forest (RF), K-Nearest Neighbors (KNN), Decision Trees (DT), Naive

Bayes (NB), Logistic Regression (LR) [20], Support Vector Machines (SVM), and Linear Discriminant Analysis (LDA); and more advanced methods such as ensembles, evolutionary computation or deep neural networks. The performance of binary classifiers is typically evaluated using metrics such as accuracy, precision, recall, and the area under the receiver operating characteristic curve (AUC-ROC). The challenge in binary classification often lies in finding the optimal decision boundary that separates the two classes while generalizing well to unseen data, particularly in high-dimensional feature spaces or when dealing with imbalanced datasets. The REIMS dataset is both high-dimensional and imbalanced, with the majority class being pairs from *different* batches of fish - the dataset has 1023 features and there are more pairs of fish from different batches than there are pairs from the same. Since the task is pair-wise instance recognition for batch detection, we must process the data such that it becomes a binary classification problem. The simplest approach is used in this paper, which concatenates the input pairs into a combined feature set with 2046 features. This is a crude approach when compared to the contrastive learning method addressed in the next section. For the imbalanced REIMS dataset, we use the weighted cross-entropy loss function to give extra priority to the correct classification of the minority class, this is discussed in further detail in section 3.4.

3.2 Siamese Networks and Contrastive Learning

Figure 1 illustrates the architecture and process flow of a Siamese network used for contrastive learning, a powerful approach for learning similarity metrics in various domains. In this architecture, two input samples are processed through an identical encoder with shared weights, transforming the inputs into an embedding space where similar pairs are brought closer together while dissimilar pairs are pushed apart. The embedded representations are then compared using a similarity measure, typically cosine similarity, as shown in the figure. The contrastive loss function, depicted at the bottom of the flowchart, quantifies the network's performance by minimizing the distance between similar pairs and maximizing the distance between dissimilar pairs, subject to a margin. This approach is particularly effective in scenarios where labelled pairs are available but direct class labels might be scarce, as it encourages the network to develop an embedding space where the distance between points is indicative of their semantic similarity.

The task of instance recognition for batches is very similar to signature verification, a pair-wise comparison that predicts if two instances have the same origin, i.e. they both belong to the same batch of fish, or the signatures match. Siamese networks [8] were originally developed for the task of signature verification. Given two signatures, an authentic signature known to belong to an individual, and the "query signature" whose veracity is being tested, determine if the query and reference signature were written by the same person. The model would predict if a signature is genuine or forged. The instance recognition task for batch detection is a simplification, a pair-wise comparison between two rapid spectrometry samples, to see if they originate from the individual batch of fish.

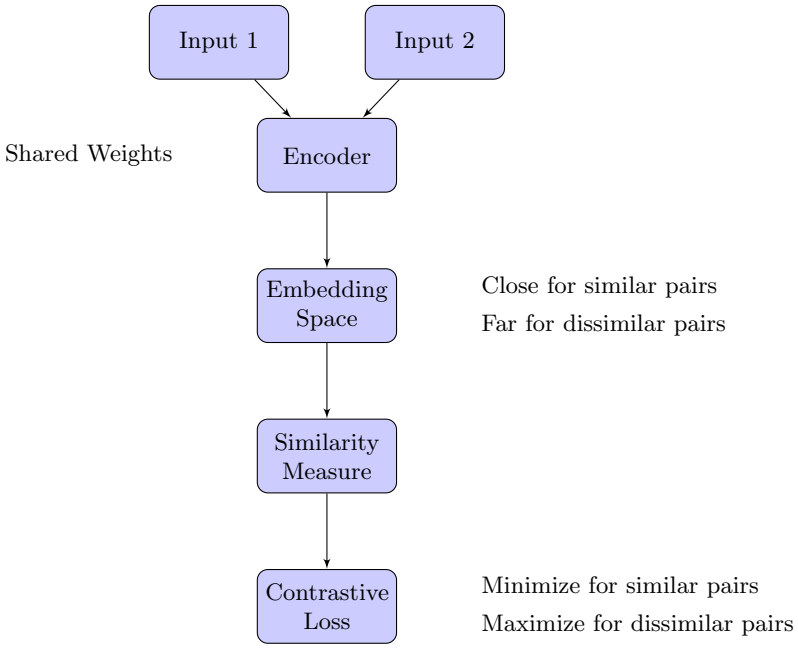

Fig. 1: Flowchart of Contrastive Learning

Although signature verification and batch detection are from different domains, the task is identical. Given two inputs, predict if they are the same. Siamese networks consist of two identical neural networks, sharing the same weights and architecture. Given a pair of signatures, a reference, and a query, one network takes the reference, the other network takes the query. The output of both networks is combined using a distance metric, to produce a similarity score. The distance metric is discussed in the following section in greater detail.

3.3 Distance Measure

Cosine similarity and Euclidean distance are both measures used to quantify the relationship between two vectors, but they capture different aspects of this relationship. In the original paper [8], the Euclidean distance was used to compute the distance between the two outputs. The score would indicate the similarity between the two signatures, the closer the Euclidean distance, the more likely the query was genuine. The greater the distance, the more likely the query was forged. Euclidean distance measures the straight-line distance between two points in Euclidean space. It is defined as:

$$d = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \tag{1}$$

Where:

– a_i and b_i are the ith components of vectors $\mathbf{A}$ and $\mathbf{B}$

Properties:

– Range: $[0, \infty)$
– 0 indicates identical vectors
– Increases as vectors become more different

However, in this paper, we use the cosine similarity to measure the distance between each pair. Cosine similarity measures the cosine of the angle between two vectors in a multi-dimensional space. It is defined as:

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} \tag{2}$$

Where:

– $\mathbf{A} \cdot \mathbf{B}$ is the dot product of vectors $\mathbf{A}$ and $\mathbf{B}$
– $\|\mathbf{A}\|$ and $\|\mathbf{B}\|$ are the magnitudes (Euclidean norms) of vectors $\mathbf{A}$ and $\mathbf{B}$

Properties:

– Range: $[-1, 1]$ for general vectors, $[0, 1]$ for non-negative vectors
– 1 indicates identical orientation
– 0 indicates orthogonality (perpendicular vectors)
– -1 indicates opposite orientation

3.4 Loss Functions

Balanced Accuracy Balanced Accuracy is a metric used to evaluate the performance of a binary classifier, especially when dealing with imbalanced datasets. It is defined as the average of sensitivity (true positive rate) and specificity (true negative rate).

$$\text{Balanced Accuracy} = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{3}$$

Where:

– TP is the number of true positives
– TN is the number of true negatives
– FP is the number of false positives
– FN is the number of false negatives

In the context of a loss function, we typically use:

$$\text{Balanced Accuracy Loss} = 1 - \text{Balanced Accuracy} \tag{4}$$

Cross Entropy Cross Entropy Loss, also known as Log Loss, measures the performance of a classification model whose output is a probability value between 0 and 1. For binary classification, it is defined as:

$$\text{CE}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{5}$$

Where:

- N is the number of samples
- y_i is the true label (0 or 1) for the i-th sample
- $\hat{y}_i$ is the predicted probability of the positive class for the i-th sample

Weighted Cross Entropy Weighted Cross Entropy is a variant of the standard cross-entropy loss that is particularly useful for dealing with imbalanced datasets. It addresses the issue of class imbalance by assigning different weights to each class, typically giving higher importance to the minority class. This approach helps prevent the model from being biased towards the majority class.

For binary classification, the Weighted Cross Entropy loss is defined as:

$$\text{WCE}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}[w_1 y_i \log(\hat{y}_i) + w_0(1 - y_i)\log(1 - \hat{y}_i)] \tag{6}$$

Where:

- N is the number of samples
- y_i is the true label (0 or 1) for the i-th sample
- $\hat{y}_i$ is the predicted probability of the positive class for the i-th sample
- w_1 is the weight for the positive class (typically the minority class)
- w_0 is the weight for the negative class (typically the majority class)

The weights w_1 and w_0 are often set inversely proportional to the class frequencies in the training data. A common approach is to use:

$$w_1 = \frac{N}{2N_1}, \quad w_0 = \frac{N}{2N_0} \tag{7}$$

Where N_1 and N_0 are the number of samples in the positive and negative classes, respectively, and $N = N_1 + N_0$ is the total number of samples.

This weighting scheme ensures that both classes contribute equally to the loss, regardless of their proportions in the dataset. As a result, the model is encouraged to pay more attention to the minority class during training, helping to mitigate the bias towards the majority class that often occurs in imbalanced datasets. By using Weighted Cross Entropy, we can effectively train models on imbalanced datasets, improving their performance across all classes and reducing the risk of overlooking minority classes that may be crucial in many real-world applications.

Contrastive Loss Contrastive Loss is used in Siamese networks to learn a similarity metric between pairs of samples. In this case, we use cosine similarity as the distance measure. The loss function encourages similar samples to have high similarity and dissimilar samples to have low similarity.

Let z_1 and z_2 be the encoded representations of two input samples, and y be the label indicating whether the pair is similar ($y = 1$) or dissimilar ($y = 0$). The cosine similarity is defined as:

$$\cos\_\text{sim}(z_1, z_2) = \frac{z_1 \cdot z_2}{\|z_1\|\|z_2\|} \tag{8}$$

The Contrastive Loss function is then defined as:

$$\begin{aligned} \text{CL}(z_1, z_2, y) =& y \cdot (1 - \cos\_\text{sim}(z_1, z_2))^2 + \\ & (1 - y) \cdot \max(0, \cos\_\text{sim}(z_1, z_2) - \text{margin})^2 \end{aligned} \tag{9}$$

Where:

- margin is a hyperparameter (often set to 0.1) that determines the minimum distance between dissimilar pairs
- cos_sim is the cosine similarity distance metric between pairs
- y is the label (1 for similar pairs, 0 for dissimilar pairs)
- z_1 and z_2 are the encoded representations of the input pairs

This loss function encourages similar pairs to have a cosine similarity close to 1, while pushing the similarity of dissimilar pairs below the margin.

4 Experimental Results

With our methodology clearly defined, we proceed to the experimental phase of our study, where we put these diverse machine learning techniques to the test on our REIMS dataset.

4.1 Dataset

The study utilizes a dataset generated through Rapid Evaporative Ionisation Mass Spectrometry (REIMS), a cutting-edge analytical technique in chemistry that allows for real-time, in-situ analysis of samples. REIMS works by rapidly heating a sample to create an aerosol of ionized molecules, which are then analyzed by a mass spectrometer, providing a detailed chemical profile of the sample. This dataset comprises 72 fish samples, originating from 24 distinct batches, with each batch contributing 6 fish. For the task of instance recognition of batches, the data is structured as pairs of fish samples, labelled to indicate whether they are from the same batch or different batches. The dataset contains 3600 instances, created by generating 50 pairs for each of the 72 fish samples. Each sample is characterized by 1023 features, presenting a high-dimensionality problem that is

susceptible to the curse of dimensionality [21]. This high feature count can lead to increased computational complexity and potential overfitting. The dataset is divided equally for training and testing purposes, with a 50%-50% split. Notably, the dataset exhibits an imbalance in the training and test set. In the training set 63.38% (1141) of the pairs represented different batches and only 36.62% (659) represented the same batch. This imbalance presents additional challenges for model training and evaluation, as it can lead to biased predictions favouring the majority class. Addressing this imbalance is crucial to ensure the model performs well across both classes, potentially requiring techniques such as oversampling, undersampling, or the use of specialized loss functions to mitigate the effects of class disproportion - in this work, we opt for the latter using contrastive loss, balanced accuracy and weighted cross entropy, these loss functions are detailed in section 3.4.

4.2 Experimental Setup

To ensure a robust and unbiased evaluation of the model's performance, we employed the balanced accuracy score as our primary metric. This choice is particularly appropriate for our imbalanced dataset, as it provides a more reliable measure of performance across all classes, regardless of their relative sizes. To establish statistical significance and account for the inherent variability in neural network training, we conducted 30 independent runs for each experiment. This approach allows for a more comprehensive assessment of the models' stability and generalization capabilities. Each of the deep learning methods has early stopping [28], effectively tuning the hyperparameter of epochs to the validation set.

In terms of loss functions, we tailored our approach to the specific learning tasks at hand. For contrastive learning, we utilized a combination of cross-entropy loss and contrastive loss, with a balancing coefficient α set to 0.5. This balanced formulation ensures that the model learns both to classify instances correctly and to distinguish between similar and dissimilar pairs effectively. In the case of binary classification, we employed the balanced classification accuracy as our loss function for the traditional machine learning methods, and weighted cross entropy for the deep learning and evolutionary computation methods, which aligns with our choice of evaluation metric and helps mitigate the effects of class imbalance during training. These loss functions play a crucial role in guiding the learning process and are discussed in greater detail in section 3.4, where we provide a thorough analysis of their mathematical foundations and practical implications. By carefully selecting these evaluation metrics and loss functions, we aim to provide a comprehensive and fair assessment of the various models' performance on our challenging dataset.

4.3 Results

Table 1 presents a comprehensive comparison of various machine learning methods across binary classification and contrastive learning tasks. The table is struc-

tured with methods listed vertically and performance metrics (train and test accuracies) for both tasks displayed horizontally. Best performing models are highlighted in bold, while second-best performers are in italics. For binary classification, Decision Trees (DT) achieve the highest training accuracy (100%), but Transformers excel in test accuracy (70.09%). In contrastive learning, LSTM shows the best training performance (98.65%), while Mamba leads in test accuracy (76.96%). Notably, there's a significant discrepancy between training and test accuracies in binary classification, particularly for traditional methods like DT and RF, suggesting overfitting. Contrastive learning appears to mitigate this issue, with deep learning models (LSTM, Mamba, Transformer) demonstrating better generalization. Genetic Programming (GP) is the only method that performs better for binary classification (65.80%) than it does for contrastive learning (61.27%). The majority of the deep learning methods outperform the traditional OPLS-DA method from the existing literature, even more so for deep learning methods combined with contrastive learning, where all methods offer superior performance. The results indicate that while traditional methods struggle with this dataset, advanced neural architectures, especially when coupled with contrastive learning, offer superior performance and generalization capabilities. The contrastive learning approach generally yields higher test accuracies compared to binary classification, suggesting its effectiveness in learning meaningful representations from the data.

Method	Binary Classification		Contrastive Learning	
	Train	Test	Train	Test
KNN	52.86% ± 1.30%	49.88% ± 0.09%		
DT	**100.0% ± 0.00%**	52.28% ± 3.32%		
LDA	57.53% ± 0.99%	54.07% ± 2.64%		
NB	63.00% ± 1.24%	53.94% ± 3.67%		
RF	*99.94% ± 0.17%*	51.48% ± 1.59%		
SVM	52.84% ± 1.27%	51.82% ± 2.04%		
LR	53.45% ± 1.17%	51.58% ± 1.53%		
OPLS-DA	57.08% ± 1.46%	53.19% ± 2.22%		
Ensemble	58.56% ± 0.99%	51.48% ± 1.59%		
CNN	50.00% ± 0.00%	50.00% ± 0.00%	94.50 ± 1.17%	70.82% ± 6.67%
GP	76.08% ± 1.65%	65.80% ± 2.88%	85.35% ± 5.32%	61.27% ± 6.37%
KAN	98.80% ± 0.58%	59.28% ± 5.23%	94.47% ± 1.60%	*75.27% ± 3.24%*
LSTM	98.74% ± 0.86%	*68.02% ± 2.09%*	**98.65% ± 0.51%**	75.05% ± 5.63%
Mamba	99.10% ± 0.85%	59.80% ± 2.98%	95.23% ± 1.25%	**76.96% ± 4.11%**
Transformer	59.04% ± 12.70%	**70.09% ± 9.89%**	*97.83% ± 0.43%*	73.79% ± 2.60%
VAE	50.10% ± 0.28%	52.89% ± 1.81%	94.62% ± 5.23%	67.83% ± 3.01%

Table 1: Binary Classification and Contrastive Learning Results

Figure 2 provides a comprehensive visualization of the performance of deep learning models on both binary classification and contrastive learning tasks for marine biomass instance recognition. The graph is organized with models listed

vertically and their corresponding performance metrics displayed horizontally, separated into four categories: binary classification (train and test) and contrastive learning (train and test). Key observations and trends:

- **Contrastive Learning Superiority**: For all deep learning methods (CNN, GP, KAN, LSTM, Mamba, Transformer, VAE), contrastive learning consistently outperforms binary classification, especially in test accuracy. This suggests that the contrastive learning approach is more effective at capturing the nuances of the data and generalizing to unseen samples
- **Overfitting in Binary Classification**: Many models, show a significant disparity between training and test accuracies in binary classification. This is evident from the large gaps between the blue (train) and orange (test) boxes, indicating severe overfitting to the training data.
- **Improved Generalization with Contrastive Learning**: The gap between training and test accuracies is generally smaller for contrastive learning compared to binary classification for deep learning models. This suggests that contrastive learning helps mitigate overfitting and improves model generalization.
- **Variability in Performance**: Some models, particularly the Transformer in binary classification, show high variability in performance, as indicated by the large whiskers and outliers. This suggests that the model's performance may be sensitive to initialization or training conditions.
- **Consistent Performance of Deep Learning Models:** In contrastive learning, deep learning models (LSTM, Mamba, KAN, Transformer) consistently achieve high test accuracies, with their boxes clustered in the upper range of the plot. This indicates their robustness and suitability for this task.
- **Poor Performance of Some Methods**: Models like CNN and VAE show relatively poor performance in binary classification, with their boxes centred around the 50% accuracy mark, suggesting they struggle with this high-dimensional data.
- **Interesting Outliers**: The CNN in binary classification shows an unusual pattern with a compressed box at 50% accuracy, suggesting it might be consistently predicting the majority class, possibly due to the class imbalance in the dataset.
- **Best Performers**: For binary classification, the Transformer achieves the highest median test accuracy. In contrastive learning, Mamba, LSTM, and KAN, in that order, show the best test performances, with their boxes positioned highest on the accuracy scale.
- **Variance in Performance**: Contrastive learning methods generally show less variance in test performance compared to their binary classification counterparts, as evidenced by the smaller box sizes and shorter whiskers.

In conclusion, fig. 2 clearly illustrates the superiority of contrastive learning approaches for deep learning architectures for instance recognition of marine biomass batch detection. It also highlights the challenges that binary classification methods face in handling this complex, high-dimensional dataset.

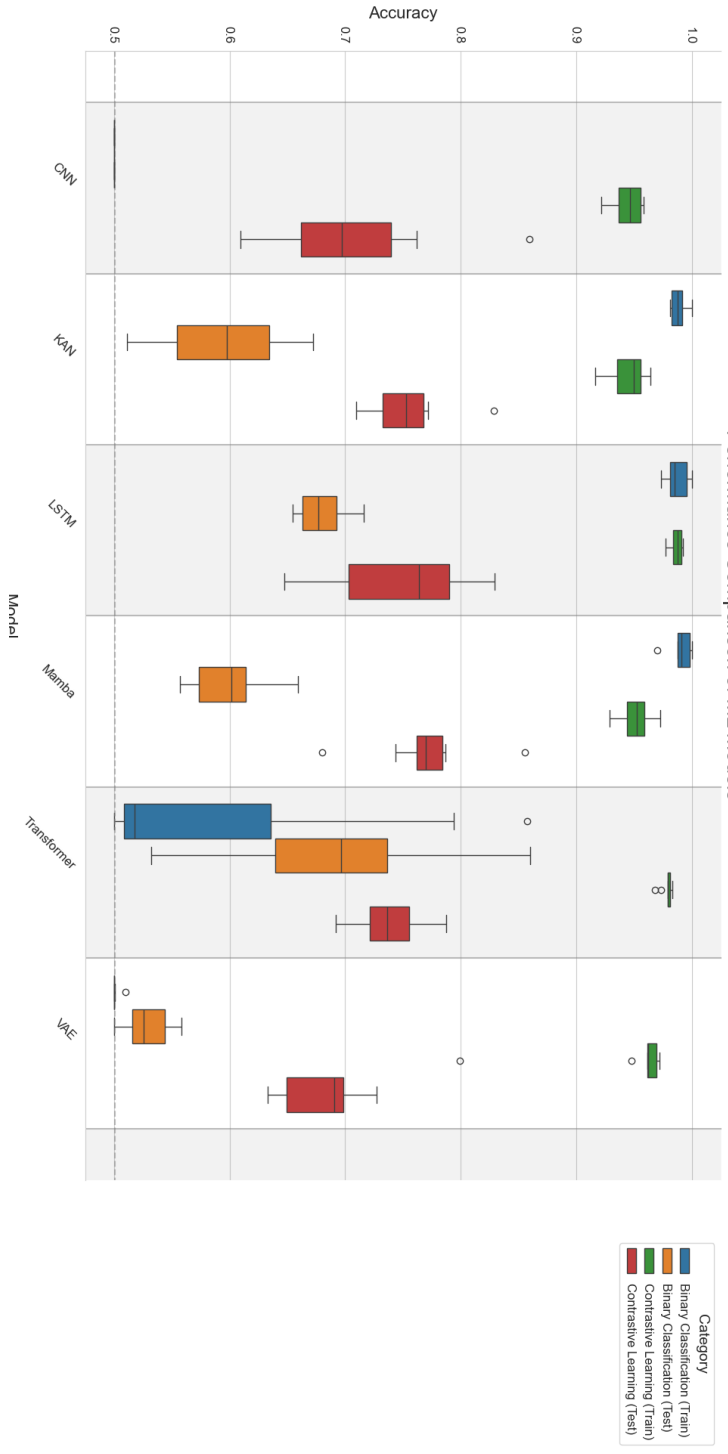

Fig. 2: Deep Learning Models for Binary Classification and Contrastive Learning Box Plot.

5 Discussions

Having presented the performance metrics of our various models, we now delve deeper into the implications of these results, focusing on the key takeaways of our findings.

The experimental results of this study on marine biomass instance recognition using REIMS data offer valuable insights into the application of various machine learning techniques. A key finding is the consistent superiority of contrastive learning approaches over traditional binary classification across all deep learning models. This suggests that the pairwise comparison framework inherent in contrastive learning is particularly well-suited to capturing subtle differences between batches of marine biomass, likely contributing to improved generalization capabilities.

Advanced deep learning architectures, notably LSTM, Mamba, and Transformers, demonstrated superior performance in both tasks, indicating their ability to extract complex, hierarchical features from high-dimensional REIMS data. However, the substantial gap between training and test accuracies, especially in binary classification, highlights the challenge of overfitting. This is likely due to the high dimensionality of the data combined with the relatively small dataset size. Contrastive learning appears to mitigate this issue to some extent, possibly by forcing the model to learn more generalizable features through pairwise comparisons. The poor performance of traditional machine learning methods underscores the complexity of the task in this high-dimensional space, as these methods may struggle to capture the intricate patterns necessary for distinguishing between batches.

The high variability observed in some models, particularly the Transformer in binary classification, suggests that performance may be sensitive to initialization and training conditions, highlighting the importance of robust evaluation methods. While not the top performer, the Genetic Programming approach showed promising results, particularly in binary classification, suggesting that evolutionary techniques could play a role in feature construction or model optimization. The improved accuracy achieved by contrastive learning methods has significant implications for quality control in marine biomass processing, potentially enabling more reliable batch identification and contamination detection.

6 Conclusion and Future Work

Our analysis of the experimental results and their implications leads us to draw several important conclusions about the efficacy of different machine learning approaches in marine biomass instance recognition, as well as identify promising avenues for future research.

This comprehensive study on instance recognition for detecting batches of marine biomass using REIMS data has yielded several important conclusions. Contrastive learning, implemented through Siamese networks, consistently outperforms traditional binary classification across various model architectures,

proving particularly effective in learning discriminative features from high-dimensional REIMS data. Advanced deep learning models, especially LSTM, Mamba, and Transformers, demonstrate superior performance in this task, indicating their ability to capture complex patterns in spectral data.

The study highlights the challenges posed by high-dimensional data in marine biomass analysis, including overfitting and the limitations of traditional machine learning methods. The results underscore the potential of machine learning, particularly contrastive learning approaches, in enhancing quality control and traceability in marine biomass processing. The comparative analysis provides valuable insights into the strengths and weaknesses of different machine learning paradigms when applied to complex recognition tasks in the domain of marine biomass analysis.

Future work could explore the integration of these models into real-time analysis systems, the application of transfer learning to leverage pre-trained models for similar tasks, and the development of interpretable AI techniques to provide insights into the specific spectral features that contribute to accurate batch identification. Additionally, investigating the performance of these models on larger and more diverse datasets could further validate their generalizability and robustness. In conclusion, this research not only advances the field of instance recognition for batch detection of marine biomass, but also provides a foundation for applying sophisticated machine learning techniques to similar challenges in analytical chemistry and quality control across various industries.

7 Acknowledgement

This work is supported in part MBIE Fund on Research Program under the contract of C11X2001. We would also like to thank our project leader Sue Marshall at Plant and Food Research.

References

1. Abdi, H., Williams, L.J.: Principal component analysis. Wiley interdisciplinary reviews: computational statistics **2**(4), 433–459 (2010)
2. Balakrishnama, S., Ganapathiraju, A.: Linear discriminant analysis-a brief tutorial. Institute for Signal and information Processing **18**(1998), 1–8 (1998)
3. Balog, J., Szaniszlo, T., Schaefer, K.C., Denes, J., Lopata, A., Godorhazy, L., Szalay, D., Balogh, L., Sasi-Szabo, L., Toth, M., et al.: Identification of biological tissues by rapid evaporative ionization mass spectrometry. Analytical chemistry **82**(17), 7343–7350 (2010)
4. Black, C., Chevallier, O.P., Cooper, K.M., Haughey, S.A., Balog, J., Takats, Z., Elliott, C.T., Cavin, C.: Rapid detection and specific identification of offals within minced beef samples utilising ambient mass spectrometry. Scientific reports **9**(1), 1–9 (2019)
5. Black, C., Chevallier, O.P., Haughey, S.A., Balog, J., Stead, S., Pringle, S.D., Riina, M.V., Martucci, F., Acutis, P.L., Morris, M., et al.: A real time metabolomic profiling approach to detecting fish fraud using rapid evaporative ionisation mass spectrometry. Metabolomics **13**(12), 1–13 (2017)

6. Boccard, J., Rutledge, D.N.: A consensus orthogonal partial least squares discriminant analysis (opls-da) strategy for multiblock omics data fusion. Analytica chimica acta **769**, 30–39 (2013)
7. Breiman, L.: Classification and regression trees. Routledge (2017)
8. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a" siamese" time delay neural network. Advances in neural information processing systems **6** (1993)
9. Bylesjö, M., Rantalainen, M., Cloarec, O., Nicholson, J.K., Holmes, E., Trygg, J.: Opls discriminant analysis: combining the strengths of pls-da and simca classification. Journal of Chemometrics: A Journal of the Chemometrics Society **20**(8-10), 341–351 (2006)
10. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3), 273–297 (1995)
11. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
12. Fix, E., Hodges, J.L.: Discriminatory analysis. nonparametric discrimination: Consistency properties. International Statistical Review/Revue Internationale de Statistique **57**(3), 238–247 (1989)
13. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752 (2023)
14. Hand, D.J., Yu, K.: Idiot's bayes—not so stupid after all? International statistical review **69**(3), 385–398 (2001)
15. Hansen, L.K., Salamon, P.: Neural network ensembles. IEEE transactions on pattern analysis and machine intelligence **12**(10), 993–1001 (1990)
16. Ho, T.K.: Random decision forests. In: Proceedings of 3rd international conference on document analysis and recognition. vol. 1, pp. 278–282. IEEE (1995)
17. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
18. Jing, L., Zhu, J., LeCun, Y.: Masked siamese convnets. arXiv preprint arXiv:2206.07700 (2022)
19. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
20. Kleinbaum, D.G., Dietz, K., Gail, M., Klein, M., Klein, M.: Logistic regression. Springer (2002)
21. Köppen, M.: The curse of dimensionality. In: 5th online world conference on soft computing in industrial applications (WSC5). vol. 1, pp. 4–8 (2000)
22. Koza, J.R., et al.: Genetic programming II, vol. 17. MIT press Cambridge (1994)
23. LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., Jackel, L.: Handwritten digit recognition with a back-propagation network. Advances in neural information processing systems **2** (1989)
24. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural computation **1**(4), 541–551 (1989)
25. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
26. LeCun, Y., et al.: Generalization and network design strategies. Connectionism in perspective **19**(143-155), 18 (1989)
27. Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T.Y., Tegmark, M.: Kan: Kolmogorov-arnold networks. arXiv preprint arXiv:2404.19756 (2024)

28. Morgan, N., Bourlard, H.: Generalization and parameter estimation in feedforward nets: Some experiments. Advances in neural information processing systems **2** (1989)
29. Tran, B., Xue, B., Zhang, M.: Genetic programming for feature construction and selection in classification on high-dimensional data. Memetic Computing **8**(1), 3–15 (2016)
30. Tran, B., Xue, B., Zhang, M.: Genetic programming for multiple-feature construction on high-dimensional classification. Pattern Recognition **93**, 404–417 (2019)
31. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
32. Zhu, J., Jang-Jaccard, J., Singh, A., Welch, I., Harith, A.S., Camtepe, S.: A few-shot meta-learning based siamese neural network using entropy features for ransomware classification. Computers & Security **117**, 102691 (2022)