



Using extended siamese networks to provide decision support in aquaculture operations

Bjørn Magnus Mathisen^{1,2} · Kerstin Bach^{1,2} · Agnar Aamodt^{1,2}

Accepted: 31 January 2021 / Published online: 26 March 2021
© The Author(s) 2021

Abstract

Aquaculture as an industry is quickly expanding. As a result, new aquaculture sites are being established at more exposed locations previously deemed unfit because they are more difficult and resource demanding to safely operate than are traditional sites. To help the industry deal with these challenges, we have developed a decision support system to support decision makers in establishing better plans and make decisions that facilitate operating these sites in an optimal manner. We propose a case-based reasoning system called aquaculture case-based reasoning (AQCBR), which is able to predict the success of an aquaculture operation at a specific site, based on previously applied and recorded cases. In particular, AQCBR is trained to learn a similarity function between recorded operational situations/cases and use the most similar case to provide explanation-by-example information for its predictions. The novelty of AQCBR is that it uses extended Siamese neural networks to learn the similarity between cases. Our extensive experimental evaluation shows that extended Siamese neural networks outperform state-of-the-art methods for similarity learning in this task, demonstrating the effectiveness and the feasibility of our approach.

Keywords Machine learning · Extended siamese neural networks · Siamese neural networks · Case-based reasoning · Decision support systems

1 Introduction

Aquaculture is currently a growing industry in Norway and is expected to grow five-fold by year 2050 [20]. As a result, the industry needs to expand to new locations that

are increasingly more exposed to harsh weather conditions, as they are located further away from the coast. As the aquaculture industry is set to grow, it also faces scrutiny related to its environmental impact, including lice growth on salmon and the deposition of waste on the seabed below aquaculture locations. Traditionally, the industry has selected locations that are sheltered in fjords or behind islands, as more exposure makes the operation of the site expensive and dangerous. However, as a result of the environmental impacts at these aquaculture locations, government agencies do not allow locations to be too closely grouped geographically. Thus, new locations for aquaculture operations may be more exposed to weather. The aquaculture industry is already the second most dangerous profession in Norway with regard to work-related accidents [12, 13]. An important task in the future is to develop new technology for the aquaculture industry that helps alleviate the increased risk resulting from the use of exposed locations. Hence, it has become a priority to reduce the amount of manual work and increase the level of automation. Decision support systems (DSSs) can help managers to plan operations at sites and minimize the amount of manual work.

This work was supported by the Research Council of Norway through the EXPOSED project (grant number 237790) and the Norwegian Open AI Lab.

✉ Bjørn Magnus Mathisen
bjornmm@ntnu.no

Kerstin Bach
kerstin.bach@ntnu.no

Agnar Aamodt
agnar@ntnu.no

¹ Department of Computer Science, Norwegian Open AI Lab, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

² EXPOSED Aquaculture Research Centre, Department of Computer Science, Trondheim, Norway

DSSs for the aquaculture industry have been developed for many years and utilize a wide range of different machine learning and artificial intelligence techniques [19]. Aquaculture is an industry that traditionally relies upon experience compiled over years. As a result, the aquaculture industry does not have a tradition of establishing methods and frameworks to create a formal framework for industry practices.

Case-based reasoning (CBR) is a machine learning method in which learning is performed by storing cases that describe previously encountered problems and their solutions. A new unseen problem can then be solved by retrieving the stored case that most closely resembles the new problem. The whole process can be described via the CBR cycle [1]: retrieve the most similar case, reuse this case to solve the new problem, revise the retrieved case if the solution is adapted for the new problem, and finally retain/store the revised case. All parts of this cycle can be adapted when designing a CBR system to solve a problem. However, a natural point of focus is the retrieval phase, in which the CBR system must calculate the similarity or distance between the problem case and all the stored cases. This step is usually performed via similarity functions, which can be modeled by domain experts or learned from data. Introducing a DSS system using the CBR methodology for the aquaculture industry can benefit from being able to capture and reuse past experiences to make predictions or recommendations produced by the system. Applying DSS in this fashion builds on the tradition in industry of drawing upon previous experiences.

In this work, we propose AQCBR, which is a DSS that supports aquaculture site operators in planning operations for their locations. The procedures in the aquaculture industry are traditionally based on extensive experience and intuition. For a DSS to be effective in such a work culture, any prediction made by the DSS should also be explained to the user. Although not the main focus of our work, CBR is an established method for explainable systems [14, 22]. AQCBR provides this same approach by using previously recorded cases as a basis for predictions. In our work, we created a case base from a dataset that was collected previously in a project named EXPOSED for studying exposed aquaculture operations¹. This case base was populated when developing AQCBR. We then adapted several similarity learning methods to AQCBR to learn a representation of the similarity between the operations. We evaluated the different similarity learning methods by quantitatively focusing on AQCBR's usefulness for classification and then qualitatively applied similarity matrices. Our work focuses on evaluating similarity learning methods, and for retrieval, we use a standard linear retrieval

method that compares all stored cases against the query case.

The main contribution of this paper is the application of extended Siamese neural networks [17] (ESNNs) to learn the similarity between cases of aquaculture operations that are stored in AQCBRs. An ESNN is then evaluated in the context of the problem of comparing aquaculture operations, and this approach is shown to outperform other state-of-the-art learning/distance similarity metric learning methods.

The paper is organized as follows. In Section 2, we outline the previous work related to machine learning applied to DSSs for aquaculture. Section 3 describes the different data sources, how the data were gathered, how the data were interpreted, how we combined the different data sources into a case and how the cases were grouped into a case-base. In Section 4, we describe the method used for learning the similarity function and the other methods used as references. Section 5 presents the results of evaluating the methods on the target task. Finally, we discuss and interpret the results of the experiments in Section 6.

To make this work reproducible, the code for the experiments described in this paper is available at <https://github.com/ntnu-ai-lab/esnn-aqcb>.

2 Related work

CBR has been applied to DSS systems in aquaculture before, such as in the work done by Tidemann et al. [24] on operational support in fish farming and in addition to previous work on CBR for predicting the success of marine operations [18]. As part of the work presented in this paper, we focus on learning a similarity function to correctly retrieve the most-similar case for supporting the aquaculture operator in planning operations.

Learning similarity functions from data reduces the work of developers and domain experts in modeling the similarity function manually when designing a CBR system. Methods for learning similarity measures have been a topic of research of the CBR community for many years [2, 23] and have also been of heightened focus recently [7, 11, 27]. Different types of methods for learning similarity have been used for many tasks, such as deep metric learning for human activity recognition [16] and ESNNs for 14 different domains [17]. Dieterle et al.'s work [6], in which the features of cases were weighted with an ANN, is another example of the application of a learned similarity function.

Siamese neural networks are a subset of a class of machine learning techniques grouped under the definition of deep metric learners (DMLs). DMLs are optimized to learn an embedding function for datapoints and then calculate the similarity or distance between two such embeddings.

¹<https://exposedaquaculture.no/>

One of the first examples of DML was Siamese neural networks (SNNs), which are trained on pairs of cases. The first example of this approach was given by Bromley et al. [4] where a SNN was used to compare signatures. This usage of DMLs follows a pattern in which DMLs are applied to problems with many classes, such as the identification of human activities [16], signatures [4] or persons [21]. Typically, a SNN learns on pairs of datapoints, and the loss function is calculated based on whether the two most similar cases fall within the same class. Recent developments in DML methods have expanded this approach to triplet networks [10], where the DML is trained on sets of triplets consisting of an anchor datapoint, a positive datapoint (same class as the anchor datapoint) and a negative datapoint (different class than the anchor case). Matching networks [26] are also a subclass of DMLs trained on representatives from clusters in the dataset. Some methods of similarity learning, such as that proposed by Gabel et al. [7], which learns the similarity of two data points based on training with concatenations of each pair of datasets, fall outside the DML class of methods.

Another application of SNNs is target tracking, i.e., tracking objects across video frames [3, 8, 9, 25], where the Siamese architecture is used to compute the distance/correlation between two image patches. In this type of SNN application, Siamese networks are typically convolutional neural networks that extract information from parts of images. Some SNNs also employ long short-term memory modules to capture patterns over time between and within frames [8, 9]. The two embeddings computed by the SNNs in tracking problems are combined using a correlation operation. The output of the correlation operation can then be used to estimate the distance between the two data points. The two embeddings computed by the SNNs in tracking problems are combined using a correlation operation. The output of the correlation operation can then be used to estimate the distance between the two data points. Typically, SNNs are fully symmetric end-to-end with regard to the two inputs; in contrast, some of the SNN methods used for visual tracking [8, 9, 25] apply an operation to only one of the signals before they are combined to calculate the distance/correlation.

In previous work, we developed ESNNs, as described in [17]. An ESNN is an extension of a SNN that has a higher capacity than a SNN for learning and differentiating between classes/categories in a metric learning task. In addition to learning embeddings in the way a SNN does, eSNNs learn how to use the differences between two such embeddings to calculate the corresponding distance. Thus, an eSNN is a Type-4 similarity function [17]; such functions have displayed excellent performance in terms of similarity

learning with datasets that are hard to classify. We use an eSNN to create a DSS based on a CBR system (AQCBR) to predict failures in operations at exposed aquaculture locations.

3 Operational situation dataset and case definition

As part of the EXPOSED project, operational data were gathered at three different aquaculture locations. Each of the locations was exposed to weather and harsh environments at a level well above average in the industry.

3.1 Reports

The operators of the sites were tasked with recording whether a set of possible operations was possible to perform that day or if the weather or environment was too challenging to safely execute these operations.

The operations considered each day were:

- Go out to the site (all of the personnel typically do not live/sleep at the location);
- Perform a daily inspection, go out on the seacage structure, and inspect the structure itself, as well as the fish, to ensure sufficient fish welfare;
- Use of a crane on boat in relation to the location structures. Typically, strong winds or waves make operating a crane from a boat very difficult because the length of the crane amplifies the movement generated by the waves on the boat;
- Use a winch from a boat to operate at the location (e.g. pull up different parts of the underwater structure);
- Operate a wellboat on location. Wellboats are used to collect fully grown fish for slaughter or deliver spawn to the fish farm cages;
- Perform delicing operations on location. Typically, delicing is performed with wellboats using either chemical, temperature or mechanical methods;
- Deliver fish food and freshwater via a feedboat equipped without dynamic positioning (DP) equipment; and
- Deliver fish food and freshwater via a feedboat equipped with DP equipment.

In addition, if any operation was deemed too difficult because of the weather or environment, each of the reports had to specify whether the limitation was related to winds, waves, currents or a combination of these factors that hindered the operation. Table 1 shows example reports from four days over two weeks. In total, 708 reports were recorded from 05.12.2016 to 30.12.2018.

Table 1 Example of two cases from different aquaculture sites from two days over a period of two calendar weeks. The features of the cases are binary, where “1” (highlighted in bold in the table) indicates a failure for that operation at that time and location

Parameter/Week number	Week 39	Week 40
Date	30.09.2017	02.10.2017
Go out to site (wind)	0	0
Go out to site (wave)	0	0
Go out to site (current)	0	0
Daily inspection (wind)	0	1
Daily inspection (wave)	0	1
Daily inspection (current)	0	0
Use of crane on boat (wind)	0	1
Use of crane on boat (wave)	0	1
Use of crane on boat (current)	0	0
Use of nokke on boat (wind)	0	0
Use of nokke on boat (wave)	0	0
Use of nokke on boat (current)	0	0
Usage of wellboat (wind)	0	1
Usage of wellboat (wave)	0	1
Usage of wellboat (current)	0	0
Usage of service boat (wind)	0	1
Usage of service boat (wave)	0	1
Usage of service boat (current)	0	0
Delicing (wind)	0	1
Delicing (wave)	0	1
Delicing (current)	0	0
Usage of feedboat w/anchor (wind)	0	0
Usage of feedboat w/anchor (wave)	0	0
Usage of feedboat w/anchor (current)	0	0
Usage of feedboat w/DP (wind)	0	0
Usage of feedboat w/DP (wave)	0	0
Usage of feedboat w/DP (current)	0	0

Conversely, “0” indicates that the operation was successful. The features of the cases have values for each operation type, indicating whether it was wind, waves or currents as the reason for failure. The example from week 39 shows a case in which all operations were successful. The second example case from week 40 shows a failure to perform daily inspection, use a crane on a boat, use a wellboat, use a service boat and perform delicing. All the failures in the case from week 40 were due to wind and waves and were not because of currents. Additionally, the usage of feedboat was still possible because these boats are large and their operations less sensitive to weather than is the operation of smaller boats

3.2 Weather reports

The Norwegian Meteorological Institute provides historical records of weather data through its API.² This API provides recorded weather data from the closest weather station to a given point in Norway. Thus, we could collect wind speed

²<https://frost.met.no>

and wind direction at the location and time for each report. However, the different weather stations and their sensors do fail from time to time, so for some days, the closest operational weather station may be farther away from the location of the aquaculture operation than it is on other days. As a result, we calculated the distance from the weather station to the location for each report as a feature in the dataset.

3.3 Exposure level and wind effect

The EXPOSED project has produced a dataset [15] that describes the degree of exposure for most of the aquaculture installations in Norway. The data set provides a level of exposure of 360 degrees around the installation. One such installation and its exposure is illustrated in Fig. 1. The exposure level is quantified in the range from 0 to 1, where 0 represents that the installation is shielded by a landmass and 1 represents no land within 40 km. This dataset provides the exposure level in the direction of the wind at any point in time.

It is intuitive to incorporate the exposure level data into the cases so that a learned similarity function can compare levels of exposure between sites when computing the similarity between operational situations. Including all 360 data points per site in every report would be counterproductive for several reasons. First, the exposure data do not change over time for each site. Second, only a small portion of the exposure data in the direction of the

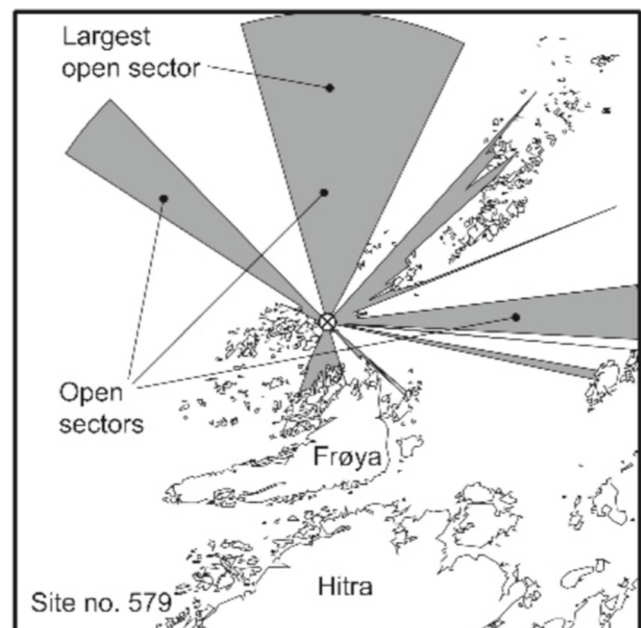


Fig. 1 Level of exposure at one of the sites [15] that provided data for our work. One can see that the site is exposed to wind and waves from the north (marked as an open sector)

wind on a particular day influence the operations for that day (being exposed in the direction of no wind has little effect). Our solution is to combine the exposure level with the wind direction at the location and the time of the report. In this way, the learned similarity function can consider the exposure level in the same direction as the wind direction at that time. This approach is implemented as a lookup function that returns the exposure level for a given wind direction. To make the function smoother in terms of the wind effect, we add a Gaussian filter. This addition will let neighboring exposure levels influence the calculated wind effect. The lookup function gf is defined as follows:

$$gf(wd, el, wis) = G(wis + 1) \bullet el(wd, wis), \quad (1)$$

where $G(\cdot)$ returns a Gaussian filter of size wis as a vector and $el(wd, wis)$ returns the exposure level in the wind direction wd as well as the wis exposure levels adjacent to the wind direction wd . Thus, the model considers the level of exposure adjacent to the wind direction and not just the single degree of influence in the direction of the wind. In our model, we set this window wis to 10, thus accounting for the exposure level within ± 5 degrees of the wind direction. This vector of wind exposure levels in and near the wind direction can then be combined with the wind speed to obtain the wind effect. This value is defined as we :

$$we(w, wd, el, wis) = gf(wd, el, wis) \cdot w, \quad (2)$$

where w is the wind speed at the site at that current time and all other function parameters are as defined in (1).

3.4 Case definition and case base population

For all eight different types of operations listed in Section 3.1, there can be four different outcomes: no failure or failure because of wind, waves or current. As a result, $4 * 8 = 32$ classes exist, which are too many classes to learn to separate from 708 data points. However, from the perspective of a DSS user in the setting of aquaculture operation planning, a general prediction of operational failure is useful. Thus, grouping the failure types and causes reduces the resolution but retains most of the utility of AQCBR as a DSS. After grouping all the failures, we can evaluate the AQCBR's ability to predict failures related to weather. Given that these operational failures seldom occur, the dataset is unbalanced, with 88% of all cases not reporting any failures. Given a failed operation, it is highly likely that higher winds from the same direction will also cause failures. Thus, it is simple to generate realistic failure cases from the existing failure cases and expand the training dataset. To generate a realistic case, we pick a random failed operation and add a small random

value to the wind speed; this is done while making sure the data point is not noise (\cdot is associated with a low wind speed, see Fig. 2). Figure 2 shows a pair plot for a subset of the case features, with the cases colored according to class (failure/success). The pair plot shows that most failure cases occur due to high wind speeds but that some occur at low wind speeds. The failure cases of the latter type are not considered during the rebalancing of the dataset.

We can now define the case base for AQCBR. Formally, the case base data are denoted as $d = x_1, x_2, \dots, x_n$, where x_i is one report containing success or failure (sf) information for an operation. Furthermore, let $el = el_1, \dots, el_n$ be the dataset of exposure levels, where el_i corresponds to the exposure level at the location of report x_i . Let $w = w_1, \dots, w_n$ be the dataset of weather reports collected for these sites, where w_i corresponds to the report x_i . These weather reports contain wind speed (ws), wind direction (wd) and distance to weather station (di) information. Thus, a case can be represented as:

$$C_i(x_i, el_i, w_i) = w_i(ws, wd, di), we(w_i(ws), w_i(wd), el_i, wis), x_i(sf), \quad (3)$$

where $we(\cdot)$ is defined by (2) and wis is the window size for the weather effect (how much the exposure levels to either side of the wind direction are taken into account). Case bases are then split into testing (querying) and training sets based on stratified cross-validation to evaluate the AQCBR method (see Section 5). Example cases following this definition are given in Table 2.

4 Extended siamese neural networks

For AQCBR to perform well, the most appropriate case must be retrieved from the case base when presented with a query. Thus, after populating the case base (Section 3.4), we need to define a similarity function that captures the connection between weather data and exposure level at a site and whether an aquaculture operation could be successful given those circumstances. This process could be done in different ways, including manual/analytical modeling. However, in this paper, we focus on the automatic learning of the similarity function using similarity learning. Notably, differences among localities and how these differences change the way weather affects operations are hard to model manually. Our approach is to learn these connections through induction by creating a machine learning model based on collected data.

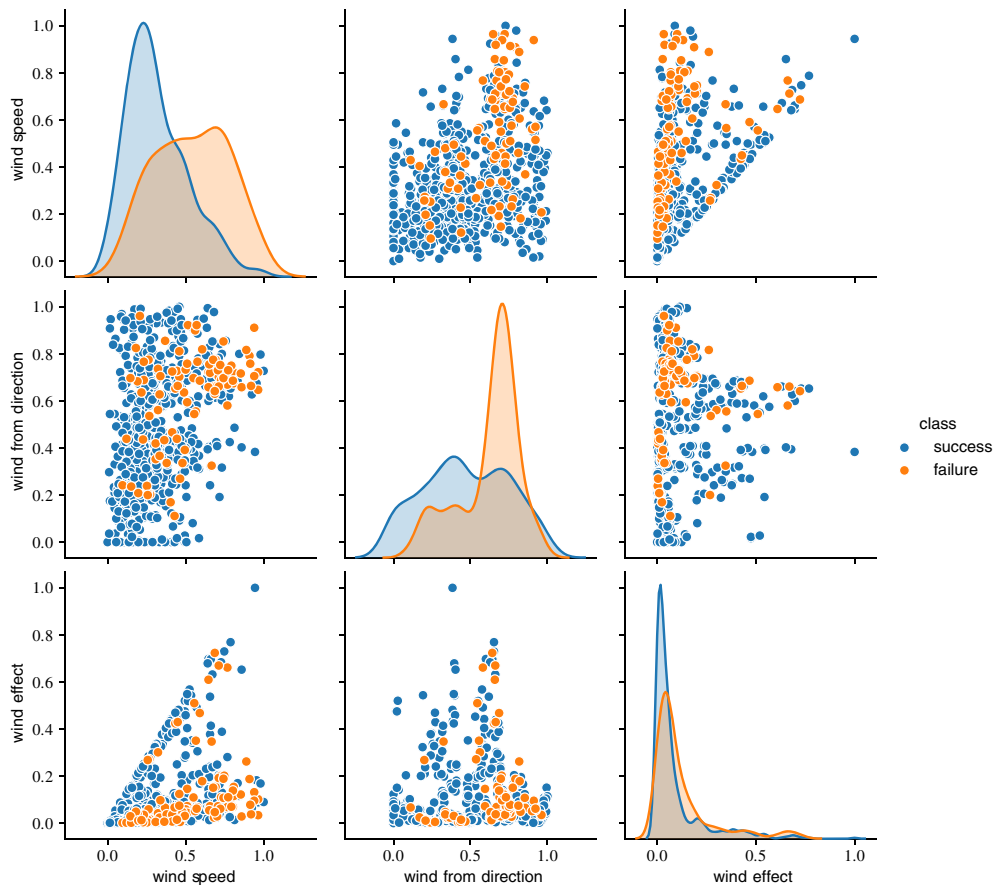


Fig. 2 Pair plot that shows the correlation among three of the features for cases from the EXPOSED dataset. The coloring of the data points denotes whether the operation was successful (blue) or failed (orange). All values were normalized (e.g., wind speed=1 is the maximum wind

speed in the dataset). We can see that the wind speed feature has two clusters according to failure or success. However, there are also some failure cases with low wind speeds

Below, we briefly describe our method of similarity learning in AQCBR and the reference methods used for comparison.

ESNNs have displayed good performance compared to other methods [17], thus an ESNN was chosen as the primary method of similarity learning for AQCBR. We refer to the implementation of an ESNN in this work as *esnn*.

For reference, we also implemented the similarity learning methods described by Chopra et al. [5] (implementation referred to as *chopra*) and Gabel et al. [7] (implementation referred to as *gabel*). *chopra* is a type-3 similarity function that learns to create useful embeddings and then calculate the L2 (Euclidian) distance between pairs of embeddings. Figure 3 shows the general architecture of the ESNN.

Table 2 This table shows two example cases from the recorded data used for the training and testing performed in this work

Example	Wind speed	Wind direction	Distance	Wind effect	Success/Failure
A	0.78 (15.6)	0.725 (261)	0.14 (8844)	0.08	failure
B	0.22 (4.5)	0.741 (267)	0.14 (8844)	0.026	success

Each example case is described with the associated wind speed, wind direction, distance between the site and weather station, the wind effect and if the case represents a success or a failure. Example A shows a failed operation with a high wind speed (15.6 m/s) from 261 degrees recorded at a weather station 8844 meters from the aquaculture location. Example B shows an operation that did not fail, with a much lower wind speed (4.5 m/s) from the same wind direction reported at the same weather station

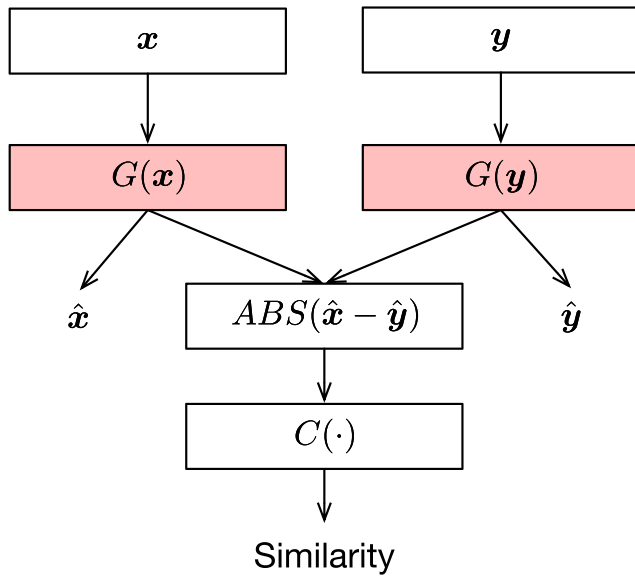


Fig. 3 Architecture of the ESNN method that provides a similarity function. This figure shows how the two datapoints \mathbf{x} and \mathbf{y} are embedded through $G(\cdot)$ (implemented by a neural network) to produce the embeddings $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$. The absolute difference vector between the two embeddings is then calculated and used as an input to a second neural network. In addition, the two embeddings are used to calculate loss associated with the classification of each of the data points

As shown in Fig. 3, *esnn* has three outputs, with two outputs used for calculating the loss ($\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$). The third output is the distance between the two data points.

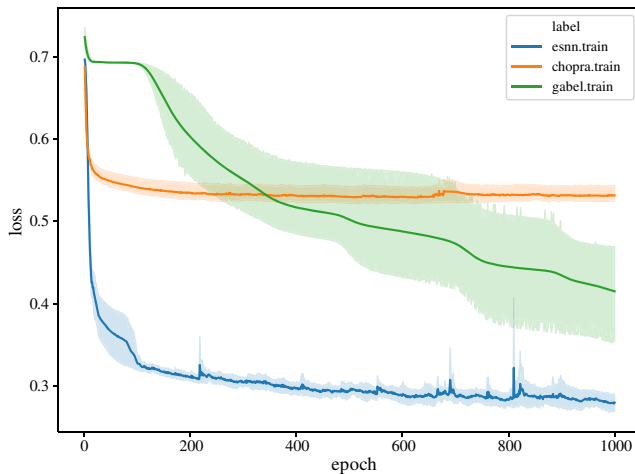


Fig. 4 Training results across 1000 epochs for the different similarity functions. The experiment was performed with five fold cross-validation and then repeated five times for validity. Similar to the results reported in [17], we can see that *esnn* and *chopra* achieve high training accuracy early in the training process, while *gabel* require additional training time as a result of its architecture. We can also observe that *chopra* reaches a performance threshold at a loss value of approximately 0.55, while *esnn* and *gabel* surpass this threshold

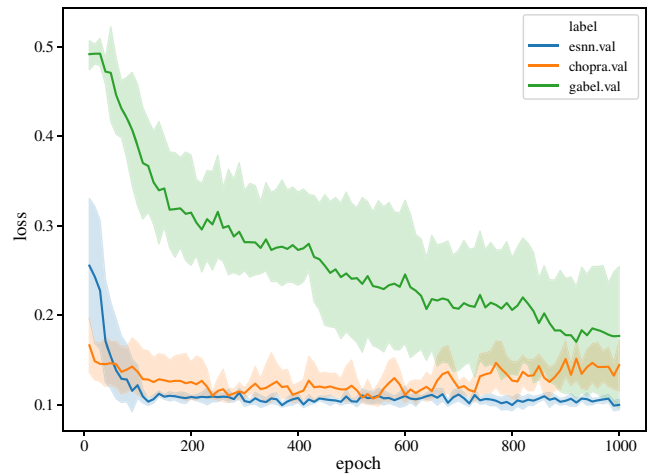


Fig. 5 Validation results across 1000 epochs for the different similarity functions. Data were recorded with the data shown in Fig. 4, with 5-fold cross validation repeated five times. Validation loss *val* was calculated every 10 epochs. This graph shows that the difference between *esnn* and *chopra* in terms of the validation accuracy is much smaller than that for the training loss, as shown in Fig. 4. As with the training performance, *gabel* takes more time to reach the same performance level, while *chopra* starts to overfit after 600 epochs

Additionally, *esnn* includes an embedding function $G(\cdot)$ and a binary function $C(\cdot, \cdot)$ that uses the two embeddings ($\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$) to compute the similarity between the two input datapoints (\mathbf{x} and \mathbf{y}). Specifically, $C(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = C(ABS(\hat{\mathbf{x}} - \hat{\mathbf{y}}))$. Let $F(\hat{\mathbf{x}})$ be the features of the case and $S(\hat{\mathbf{x}})$ the solution (or target) of case \mathbf{x} . For a pair of two cases (\mathbf{x}, \mathbf{y}) the loss function of the ESNN can then be defined as:

$$L(\alpha, \mathbf{x}, \mathbf{y}, s) = \frac{(1-\alpha)}{2} \cdot (L_c(G(\mathbf{x}), S(\mathbf{x})) + L_c(G(\mathbf{y}), S(\mathbf{y}))) + \alpha \cdot L_s(\mathbf{x}, \mathbf{y}, s), \tag{4}$$

where α is a parameter used to weight the importance of the three different outputs of the loss function and s is the true similarity. $L_c(p, q)$ is the categorical cross-entropy loss between p and q . Finally, L_s is the similarity loss, or the difference between the model-predicted similarity and the true similarity, which is defined as:

$$L_s(\mathbf{x}, \mathbf{y}, s) = |s - C(G(\mathbf{x}), G(\mathbf{y}))| \tag{5}$$

The loss function for *chopra* is $L_s(\mathbf{x}, \mathbf{y}, s)$, with $C(G(\mathbf{x}), G(\mathbf{y})) = |G(\mathbf{x}) - G(\mathbf{y})|$, and $C(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is modeled as the L2-distance between the embeddings $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$. *gabel* uses the same loss $L_s(\mathbf{x}, \mathbf{y}, s)$, but because *gabel* does not learn embeddings, $G(\cdot)$ becomes the identity function $I(\cdot)$, and $C(I(\mathbf{x}), I(\mathbf{y}))$ is learned as a neural network model.

Early experiments showed that for the case base defined in the previous section, a high α produces the best results in the similarity function for AQCBR. As a result, an

α equal to 1 was chosen, as shown in (4), resulting in $L(\alpha = 1, \mathbf{x}, \mathbf{y}, s) = L_s(\mathbf{x}, \mathbf{y})$. Thus, in our experiments, all loss functions were identical to effectively assess model performance.

5 Evaluation

The results shown in Figs. 4 and 5 were generated with five-fold stratified cross-validation and repeated five times resulting in a mean and standard deviation for each epoch. The embedding part $G(\cdot)$ of the similarity functions *esn*n and *chopra* was implemented as a fully connected ANN with three layers of size 40, 6 and 3. *esn*n had an additional two fully connected layers of size 4 and 2 to learn the binary function $C(\cdot, \cdot)$. The results in Figs. 4 and 5 show that the *esn*n similarity learning method performed better than *chopra* and *gabel*. A retrieval validation was run every 10th epoch, where every data point from the test

set of that fold was used as a query case. The training set was used as the case base. The validation performance was then calculated based on whether the most-similar case had the same solution as the query case. Figures 4 and 5 show the training and validation performance results across 1000 epochs. The training performance shows that *esn*n outperforms the reference methods *chopra* and *gabel* based on the training speed and accuracy, as well as having a slightly better validation accuracy. In addition, one can observe that *chopra* achieves effective performance very quickly, while the performance level of *gabel* gradually increases.

After 1000 epochs of training, the retrieval performance (measured as described in the previous paragraph) is 90% ($\pm 0.7\%$) for *esn*n, 85.57% ($\pm 3.4\%$) for *chopra* and 82.32% ($\pm 8.7\%$) for *gabel*.

To illustrate the qualitative results of retrieval for each of the methods, we generated similarity matrices. Figures 6, 7 and 8 show the similarity matrices and retrieval results

Fig. 6 Similarity matrix for *esn*n. The matrix is symmetric, which is a result of the architecture of the ESNN method being based on a SNN. The retrieved cases are all correctly retrieved. However, *esn*n still seems to output some similarity between the different classes. *esn*n measures *opsitu7* to be not entirely different from *opsitu1*, 2 and 3 even if they are of a different class. This result is in contrast to the similarity matrix for *chopra* shown in Fig. 8

	opsitu1	opsitu2	opsitu3	opsitu4	opsitu5	opsitu6	opsitu7	opsitu8	opsitu9	opsitu10
opsitu1	1	0.059	1	1	0.0027	0	0.2	0.033	0.13	0.038
opsitu2	0.059	1	0.059	0.059	0.89	0.89	0.67	0.81	0.82	0.91
opsitu3	1	0.059	1	1	0.0029	0.00017	0.2	0.033	0.13	0.039
opsitu4	1	0.059	1	1	0.003	0.00025	0.2	0.034	0.13	0.039
opsitu5	0.0027	0.89	0.0029	0.003	1	1	0.78	0.92	0.67	0.94
opsitu6	0	0.89	0.00017	0.00025	1	1	0.78	0.92	0.68	0.94
opsitu7	0.2	0.67	0.2	0.2	0.78	0.78	1	0.71	0.67	0.81
opsitu8	0.033	0.81	0.033	0.034	0.92	0.92	0.71	1	0.87	0.93
opsitu9	0.13	0.82	0.13	0.13	0.67	0.68	0.67	0.87	1	0.59
opsitu10	0.038	0.91	0.039	0.039	0.94	0.94	0.81	0.93	0.59	1
	1	0	1	1	0	0	0	0	0	0
	opsitu3	opsitu10	opsitu4	opsitu3	opsitu6	opsitu5	opsitu10	opsitu10	opsitu8	opsitu5

for *esnn*, *gabel* and *chopra*, respectively. These figures were generated by sampling ten random cases. The cells at the intersections of each case is the calculated similarity between the cases, where 1 is high similarity and 0 is low similarity. The second to last row of each figure shows the class (failure/dark or success/light) of the cases in each column. The colors of the cells also illustrate the degrees of similarity, with high similarity (1) denoted with dark blue and low similarity (0) shown in a light color. The last row shows the name of the most similar case (except itself) to the case in the column, and the color of the cell indicates if this case was correctly (dark) retrieved.

Figures 6 and 8 show that both *esnn* and *chopra* perform well as a similarity measures in AQCBR and generally retrieve the correct cases. Figure 7 shows that *gabel* performs slightly worse in practice. The figure also shows that *gabel* is not a symmetric similarity measure; notably, the diagonal is not a static value, and similarities change when the ordering of cases changes $gabel(opsitu2, opsitu3) \neq gabel(opsitu3, opsitu2)$.

One can also observe that *chopra* has a less smooth similarity matrix than those of *esnn* and *gabel*. *chopra* seems to clearly treat a query case as part of a class (success or failure), with near-zero similarity with cases in different classes than the query class in the case base; this is not the case with *esnn* and *gabel*, which output various similarities between a query case in one class and other cases in different classes. This feature is likely the result of how *chopra* implements its binary function $C(\cdot, \cdot)$ as a static L2 distance function. As a result, *chopra* approximates a threshold function after training. This finding suggests that similarity learning methods that learn the binary part of similarity functions (denoted as $C(\cdot, \cdot)$ in Section 4) can provide more insight into similarities between cases of different classes than can similarity functions that do not learn the binary part. In the context of AQCBR, this suggests *esnn* and *gabel* could help users obtain insight into which parts of failed operations are similar to successful operations.

Fig. 7 Similarity matrix for *gabel*. The matrix illustrates that this similarity learning method is not based on a SNN because it is not symmetric and the diagonal is not 1. In agreement with *esnn*, this method measures the failure case *opsitu7* to be close to the success cases *opsitu1*, 2 and 3. In contrast to the matrix produced by *esnn*, *opsitu7* is measured to be closer to *opsitu1*, which of a different class than *opsitu7*, than *opsitu2* which of the same class

	opsitu1	opsitu2	opsitu3	opsitu4	opsitu5	opsitu6	opsitu7	opsitu8	opsitu9	opsitu10
opsitu1	0.96	0.022	0.96	0.95	0.0052	0.0098	0.46	0.018	0.024	0.0066
opsitu2	0.019	0.99	0.015	0.014	1	1	0.31	1	0.99	0.99
opsitu3	0.96	0.019	0.95	0.95	0.005	0.013	0.38	0.016	0.023	0.0065
opsitu4	0.95	0.018	0.95	0.94	0.005	0.015	0.34	0.015	0.023	0.0065
opsitu5	9.9e-05	1	0	6.7e-05	1	1	0.34	1	0.99	1
opsitu6	0.0038	1	0.0066	0.0085	1	1	0.97	1	1	1
opsitu7	0.46	0.37	0.39	0.35	0.32	0.97	0.068	0.64	0.3	0.17
opsitu8	0.015	1	0.013	0.012	1	1	0.59	1	0.99	1
opsitu9	0.021	0.99	0.02	0.02	0.99	1	0.33	0.99	0.99	0.99
opsitu10	0.0024	0.99	0.0025	0.0027	1	1	0.19	1	0.99	0.99
	1	0	1	1	0	0	0	0	0	0
	opsitu3	opsitu6	opsitu1	opsitu1	opsitu6	opsitu5	opsitu6	opsitu6	opsitu6	opsitu6

Fig. 8 Similarity matrix for *chopra*. This similarity matrix is very binary in comparison with those from *gabel* and *esnn*, and similarity above 0.1 is only output between cases of the same class

	opsitu1	opsitu2	opsitu3	opsitu4	opsitu5	opsitu6	opsitu7	opsitu8	opsitu9	opsitu10
opsitu1	1	0.00047	1	1	0.0018	0.0059	0.062	0	0.016	0.0032
opsitu2	0.00047	1	0.0005	0.00051	1	0.99	0.94	1	0.98	1
opsitu3	1	0.0005	1	1	0.0018	0.0059	0.062	2.9e-05	0.016	0.0033
opsitu4	1	0.00051	1	1	0.0018	0.0059	0.062	4.6e-05	0.016	0.0033
opsitu5	0.0018	1	0.0018	0.0018	1	1	0.94	1	0.99	1
opsitu6	0.0059	0.99	0.0059	0.0059	1	1	0.94	0.99	0.99	1
opsitu7	0.062	0.94	0.062	0.062	0.94	0.94	1	0.94	0.95	0.94
opsitu8	0	1	2.9e-05	4.6e-05	1	0.99	0.94	1	0.98	1
opsitu9	0.016	0.98	0.016	0.016	0.99	0.99	0.95	0.98	1	0.99
opsitu10	0.0032	1	0.0033	0.0033	1	1	0.94	1	0.99	1
	1	0	1	1	0	0	0	0	0	0
	opsitu3	opsitu8	opsitu4	opsitu3	opsitu2	opsitu10	opsitu9	opsitu2	opsitu6	opsitu5

6 Conclusions and future work

In this work, we have shown the need for decision support tools to help strengthen the aquaculture industry and increase the level of automation and planning. To this end, we have presented AQCBR, a DSS based on CBR that uses a novel ESNN method for learning similarities.

Our results show that the similarity learning method *esnn* outperforms *gabel* and *chopra*. This finding is consistent with the previously reported results from Mathisen et al. [17]. In addition, the similarity matrices and retrieval results in Figs. 6 and 8 show that AQCBR performs well in terms of retrieving previous cases with the same outcome as the query cases. These figures also confirm that all three methods of similarity learning perform well, with *esnn* giving the best overall performance. We also showed that *esnn* performs well with relatively little data (a total of 708 data points).

In this paper, we have shown that AQCBR can serve as a decision support system for aquaculture operators because it not only differentiates feasible from unfeasible operations, but also provides explanation-by-example.

An extension to this work would be to not group all the different types of operations into one, as described in Section 3.1. However, this approach would require more data points to allow *esnn* to correctly separate additional categories of operational successes and failures. Because waves typically build over days, especially if the location is very exposed, including a time series of weather in relation to a location could improve the accuracy of the proposed method.

Acknowledgments We would like to thank the EXPOSED SFI project (Research Council of Norway - grant number 237790) and Norwegian Open AI Lab for their support of this work. We also thank Helge Langseth, Gunnar Senneset and Hans Vanhauwaert Bjelland for their excellent support of our work.

Funding Open access funding provided by NTNU Norwegian University of Science and Technology (incl St. Olavs Hospital - Trondheim University Hospital).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate

if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Aamodt A, Plaza E (1994) Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Commun* 7(1):39–59
- Aha DW (1991) Case-based learning algorithms. In: Proceedings of the 1991 DARPA case-based reasoning workshop, vol 1, pp 147–158
- Bertinetto L, Valmadre J, Henriques JF, Vedaldi A, Torr PH (2016) Fully-convolutional siamese networks for object tracking. In: European conference on computer vision, Springer, pp 850–865
- Bromley J, Guyon I, LeCun Y, Säckinger E., Shah R (1994) Signature verification using a “siamese” time delay neural network. In: Advances in neural information processing systems, pp 737–744
- Chopra S, Hadsell R, LeCun Y (2005) Learning a similarity metric discriminatively, with application to face verification. In: computer society conference on Computer vision and pattern recognition, 2005. CVPR 2005. IEEE vol 1, IEEE, pp 539–546
- Dieterle S, Bergmann R (2014) A hybrid cbr-ann approach to the appraisal of internet domain names. In: International conference on case-based reasoning, Springer, pp 95–109
- Gabel T, Godehardt E (2015) Top-down induction of similarity measures using similarity clouds. In: Hüllermeier E, Minor M (eds) Case-based reasoning research and development. Springer International Publishing, Cham, pp 149–164
- Gao P, Yuan R, Wang F, Xiao L, Fujita H, Zhang Y (2020) Siamese attentional keypoint network for high performance visual tracking. *Knowledge-Based Systems* 193:105448. <https://doi.org/10.1016/j.knsys.2019.105448>. <http://www.sciencedirect.com/science/article/pii/S0950705119306665>
- Gao P, Zhang Q, Wang F, Xiao L, Fujita H, Zhang Y (2020) Learning reinforced attentional representation for end-to-end visual tracking. *Inform Sci* 517:52–67. <https://doi.org/10.1016/j.ins.2019.12.084>. <http://www.sciencedirect.com/science/article/pii/S0020025519312095>
- Hoffer E, Ailon N (2015) Deep metric learning using triplet network. In: International workshop on similarity-based pattern recognition, Springer, pp 84–92
- Hoffmann M, Malburg L, Klein P, Bergmann R (2020) Using siamese graph neural networks for similarity-based retrieval in process-oriented case-based reasoning. In: case-Based Reasoning Research and development: 28th International Conference, ICCBR 2020, Salamanca, Spain, June 8-12, 2020, Proceedings. Springer. Accepted for publication
- Holen SM, Utne IB, Holmen IM, Aasjord H (2018) Occupational safety in aquaculture—part 1: Injuries in norway. *Mar Policy* 96:184–192
- Holen SM, Utne IB, Holmen IM, Aasjord H (2018) Occupational safety in aquaculture—part 2: Fatalities in Norway 1982–2015. *Mar Policy* 96:193–199
- Keane MT, Kenny EM (2019) How case-based reasoning explains neural networks: a theoretical analysis of xai using post-hoc explanation-by-example from a survey of ann-cbr twin-systems. In: Bach K., Marling C. (eds) Case-based reasoning research and development. Springer International Publishing, Cham, pp 155–171
- Lader P, Kristiansen D, Alver M, Bjelland HV, Myrhaug D (2017) Classification of aquaculture locations in Norway with respect to wind wave exposure. In: Proceedings of the ASME 2017 36th international conference on ocean, offshore and arctic engineering OMAE2017
- Martin K, Wijekoon A, Wiratunga N (2019) Human activity recognition with deep metric learners. In: Workshops Proceedings for the Twenty-seventh international conference on case-based reasoning. CEUR Workshop Proceedings, CEUR Workshop Proceedings
- Mathisen BM, Aamodt A, Bach K, Langseth H (2019) Learning similarity measures from data *Progress in Artificial Intelligence*. <https://doi.org/10.1007/s13748-019-00201-2>
- Mathisen BM, Aamodt A, Langseth H (2017) Data driven case base construction for prediction of success of marine operations, CEUR Workshop Proceedings, pp 102–111. <https://www.idi.ntnu.no/bjormmm/iccbr2017-ws.pdf>
- Mathisen BM, Haro P, Hanssen B, Björk S., Walderhaug S (2016) Decision support systems in fisheries and aquaculture: A systematic review. arXiv preprint arXiv:1611.08374
- Olafsen T, Winther U, Olsen Y, Skjermo J (2012) Value created from productive oceans in 2050. SINTEF Fisheries and Aquaculture p 83. https://www.sintef.no/globalassets/upload/fiskeri_og_havbruk/publikasjoner/verdiskaping-basert-pa-produktive-hav-i-2050.pdf
- Schroff F, Kalenichenko D, Philbin J (2015) Facenet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 815–823
- Sørmo F, Cassens J, Aamodt A (2005) Explanation in case-based reasoning—perspectives and goals. *Artif Intell Rev* 24(2):109–143
- Stahl A (2005) Learning similarity measures: a formal view based on a generalized cbr model. In: International conference on case-based reasoning, Springer, pp 507–521
- Tidemann A, Bjørnson FO, Aamodt A (2012) Operational support in fish farming through case-based reasoning. In: Advanced research in applied artificial intelligence, Springer, pp 104–113
- Valmadre J, Bertinetto L, Henriques J, Vedaldi A, Torr PH (2017) End-to-end representation learning for correlation filter based tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2805–2813
- Vinyals O, Blundell C, Lillicrap T, Wierstra D et al (2016) Matching networks for one shot learning. In: Advances in neural information processing systems, pp 3630–3638
- Ye X, Leake D, Huibregtse W, Dalkilic M (2020) Applying Class-to-Class Siamese Networks to Explain Classifications with Supportive and Contrastive Cases. In: case-Based Reasoning Research and development: 28th International Conference, ICCBR 2020, Salamanca, Spain, June 8-12, 2020, Proceedings. Springer. Accepted for publication

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Bjørn Magnus Mathisen received his M.Sc. degree in Artificial Intelligence and machine learning from Norwegian University of Science and Technology (NTNU), Norway. He is currently a PhD student at NTNU as part of Norwegian Open AI Lab and the EXPOSED Aquaculture Research Centre. He also works as a researcher at SINTEF research institute. His main interests are similarity or contrastive learning and machine learning based

decision support systems.



Agnar Aamodt is Professor Emeritus at the Norwegian University of Science and Technology (NTNU), Department of Computer Science. He holds a M.Sc. degree in Electrical Engineering from 1975, and a PhD in Computer Science from 1991. His research focus is on support for human problem solving and learning through automated data analysis and reuse of human experiences. He combines methods within case-based reasoning with other knowledge-based

methods and machine learning.



Kerstin Bach received her M.Sc. degree in Information Management and Information Technology (2007) and the Dr. rer. nat. in Computer Science (2012) from the University in Hildesheim, Germany. She is an associate professor in the Department of Computer Science at NTNU and a member of the Norwegian Open AI Lab. Her main research interests are case-based reasoning, intelligent decision support systems, explainable and human-centered AI.