This journal provides immediate open access to its content under the Creative Commons BY 4.0 license. Authors who publish with this journal retain all copyrights and agree to the terms of the above-mentioned CC BY 4.0 license

DOI: 10.2478/seeur-2023-0030

ANOMALY DETECTION ON UNIVARIATE SENSING TIME SERIES DATA FOR SMART AQUACULTURE USING DEEP LEARNING

Aleksandar Petkovski
Ph.D. candidate, Faculty of Contemporary Sciences and Technologies
South East European University, North Macedonia
a.petkovski@seeu.edu.mk

Visar Shehu
Full Professor, Faculty of Contemporary Sciences and Technologies
South East European University, North Macedonia
v.shehu@seeu.edu.mk

ABSTRACT

Aquaculture plays a significant role in both economic development and food production. Maintaining an ecological environment with good water quality is essential to ensure the production efficiency and quality of aquaculture. Effective management of water quality can prevent abnormal conditions and contribute significantly to food security. Detecting anomalies in the aquaculture environment is crucial to ensure that the environment is maintained correctly to meet healthy and proper requirements for fish farming. This article focuses on the use of deep learning techniques to detect anomalies in water quality data in the aquaculture environment. Four deep learning anomaly detection techniques, including Autoencoder, Variational Autoencoder, Long-Short Term Memory Autoencoder, and Spectral-Residual Convolutional Neural Network, were analysed using multiple real-world sensor datasets collected from IoT aquaculture systems. Extensive experiments were conducted for temperature, dissolved oxygen, and pH parameters, and the evaluation analysis revealed that the Long-Short Term Memory Autoencoder anomaly detection method showed promising results in detecting anomalies for the temperature and oxygen datasets, while the Spectral-

Residual Convolutional Neural Network demonstrated the best performance on the pH datasets.

Key words: Anomaly Detection, Deep Learning, Aquaculture, LSTM-AE, Autoencoder, Variational Autoencoder, SR-CNN.

INTRODUCTION

Aquaculture is the farming of fish, water plants, and other organisms and is one of the fastest growing food production sectors in the world. It has become a vital sector in ensuring global food security, particularly as the world population continues to grow. Maintaining optimal water conditions in aquaculture systems is essential for the growth and survival of aquatic animals. Abnormal water conditions, such as fluctuations in temperature, pH, and dissolved oxygen, can have severe impacts on the health and welfare of aquatic animals. Therefore, the detection of these abnormal conditions is crucial to minimize their negative impacts. Water quality control plays a vital part in aquaculture management, of which effective monitoring and detection of anomalies ensure that the aquaculture environment is maintained property to meet healthy and proper requirements for fish farming. With the continuous development of science and technology, intelligence and informatization in aquaculture has become a new trend (Encinas et al. 2017).

The quality of water plays a vital role in successful aquaculture management, as it directly affects the growth and health of fish. With the advancement of technology, the use of IoT (Internet of Things) in aquaculture has become increasingly popular. The usage of IoT technology in aquaculture automates collecting different types of data like temperature, pH, dissolved oxygen, salinity, ammonia, etc. These data, impact the physical, chemical, and biological characteristics of water, which are crucial for ensuring optimal conditions for aquatic organisms and avoiding any undesirable effects that could negatively impact production. For example, the temperature needs to be regulated within the suitable range to limit disease and oxygen consumption. Fish also require sufficient dissolved oxygen in the water to survive, and inadequate levels can result in slower growth and even death. The pH level of water affects the productivity and stress levels of fish, and monitoring these parameters is essential for maintaining good water quality. Using different types of sensors, the farmers can monitor the parameters of water quality and make informed decisions about water management, thereby reducing the likelihood of undesirable conditions that could harm fish health and the

environment. By analysing and predicting trends in the data, farmers can improve the quality of their products (Dupont et al. 2018).

As the use of IoT in aquaculture continues to grow, the need for reliable and accurate data analysis techniques becomes more critical. While IoT sensors can provide valuable data for farmers to optimize their aquaculture systems, the amounts of data can be overwhelming, and it can be challenging to identify the anomalies or outliers. The anomalies can occur when the device fails to self-diagnose itself and continues to function even though its sensors are malfunctioning, which can affect the IoT system and lead to a failure. Therefore, it is of vital importance to use sensors' outlier detection techniques to detect any abnormal behaviour. In real-world data there are values or records that are different to all others records and are known as anomalies or outliers (Gaddam et al. 2019). One of the most popular definitions for anomaly defined by (Hawkins 1980) is:" An observation which deviates so significantly from other observations as to arouse suspicion that it was generated by a different mechanism". The different types of anomalies can be categorized as:

- **Point anomaly**: The data points are considered as point anomalies in time-series data which deviate largely from the previous data points.
- **Contextual anomalies**: The data points are considered as contextual anomalies if those values deviate from the expected patterns within the time-series data.
- **Collective anomalies**: A subset of data points is considered as collective anomalies if those records as collection deviate significantly from the entire dataset.

With the growing importance of IoT systems, anomaly detection has become a critical task in ensuring their reliability and performance. Anomaly detection is used for solving different problems in IoT networks, like detecting anomalies in variety of industrial processes from such as detection in residential heating ventilation and air conditioning, quality control techniques in manufacturing, and detection in sensor readings from engine-based machines. Also, detection of anomalies is commonly used in identifying faults in power grid, anomalous road conditions in smart city, and anomalous behaviour in health services. Anomaly detection is primarily used to find problems or noise in data and is an important content in information science (Cook et al. 2020). Nowadays aquaculture utilizes numerous sensors to monitor water quality. Detecting unexpected patterns in sensor data is a significant challenge in aquaculture. Anomaly detection has traditionally relied on the Delphin method of fixed threshold, a technique that identifies anomalies when sensor values surpass predetermined thresholds.

However, this method has a limitation in that the thresholds are set at fixed levels and therefore it cannot detect anomalies at their early stages, which leads to poor timeliness (Liu et al. 2020).

Anomaly detection, which involves identifying unexpected events or items, has gained significant interest among researchers and practitioners. This area has been studied across various application domains and has seen significant advancements. In recent years, deep learning, a type of machine learning algorithm that is modelled after the human brain, has gained attention. These methods are able to learn complex data dynamics without making assumptions about underlying patterns, making them a highly appealing option for time-series analysis (Choi et al. 2021). The objective of this study is to detect anomalies in aquaculture time-series sensor data using deep learning techniques, specifically Autoencoder (AE), Variational Autoencoder (VAE), Long-Short Term Memory Autoencoder (LSTM-AE), and Spectral-Residual Convolutional Neural Network (SR-CNN). These techniques were employed in order to improve the reliability and scalability of aquaculture systems, ease the responsibility of administrator in maintenance, and finally reach water quality control in an intelligent manner.

The remainder of the paper is organized as follows. Section II explains the related work. Section III describes the research methodology, including data preprocessing, explanation of the deep learning algorithms, and evaluation techniques. In Section IV the research results are presented, and Section V provides the conclusion of the study.

RELATED LITERATURE

Anomaly detection in time-series data has been a significant area of research for a considerable time. Anomaly detection over univariate streams relies upon the comparison of the current observation against the local or global history of the time-series being analysed. Univariate time series represent the data output from a single source linked with the time of the observation (Cook et al. 2020). In aquaculture systems, various water quality parameters are measured, including pH, temperature, dissolved oxygen, turbidity, salinity, chloride, nitrate, and carbonates (Petkovski et al. 2021). Each of these sensors' data can be used for univariate modelling. Over the past several years, various techniques have been developed to detect anomalies in univariate time-series data, and deep learning approaches have gained a significant amount of attention because of their incredible capability to resolve these problems (Choi et al. 2021).

In our previous study (Petkovski & Shehu 2023) several machine learning algorithms were used for anomaly detection over the same aquaculture datasets. The employed machine learning algorithms were K-Means, Local Outlier Factor, and Isolation Forest. It was concluded that the Isolation Forest and K-Means clustering were effective in detecting anomalies in aquaculture univariate datasets, whereas Local Outlier Factor showed less effective results.

(Ren et al. 2019) proposed an accurate, efficient, and general approach for time-series anomaly detection by combining the Spectral Residual (SR) model with Convolutional Neural Network. The SR model was originally developed for visual saliency detection and adapted for time-series anomaly detection. Experimental results showed that the proposed approach outperformed state-of-the-art baselines on public datasets as well as Microsoft production data. Over 200 Microsoft teams, including Microsoft Bing, Office, and Azure, have used this univariate time series anomaly detection service. The SR-CNN model is highly efficient, capable of detecting anomalies in 4 million time series per minute.

(Liu et al. 2020) conducted a study on applying various deep learning-based anomaly detection methods to vertical plant wall systems for automating indoor climate monitoring and predictive maintenance. The methods included AE, LSTM, bidirectional LSTM, LSTM with multi-layer, LSTM encoder-decoder, and LSTM-ED with multi-layer. The study focused on two categories of anomalies - point anomalies and contextual anomalies. The researchers investigated and applied prediction-based and pattern recognition-based methods for anomaly detection in indoor climate. The study results indicated that the autoencoder and the long short-term memory encoder decoder model outperformed other models in detecting point anomalies and contextual anomalies, respectively. These findings indicate that these models can be used in industrial practice for monitoring and maintaining the indoor climate of vertical plant wall systems.

(Xu et al. 2018) conducted a study on unsupervised anomaly detection for seasonal KPIs in web applications, specifically for page views, number of online users, and number of orders. The proposed algorithm, Donut, was based on VAE and designed to handle local variations in the data. Results showed that Donut outperformed state-of-the-art supervised and VAE-based anomaly detection algorithms. The study suggests that anomaly detection using dimension reduction should consider reconstruction, while anomaly detection with generative models should be trained with both normal and abnormal data.

In (Braei & Wagner 2020) study, a comparison was made between multiple deep learning techniques and statistical and machine learning techniques on multiple Yahoo datasets. The deep learning techniques used were Multiple Layer Perceptron, Convolutional Neural Networks, Residual Neural Network, WaveNet, LSTM network, Gated recurrent unit, and autoencoder. The study results revealed that deep learning methods generally fall short of statistical methods in achieving accuracy values for univariate time-series benchmarks consisting only of point and collective anomalies. However, neural networks could surpass statistical methods in performance if the dataset primarily contained contextual anomalies. Although deep learning methods require more computational time compared to statistical approaches, some of them have been found to outperform the latter by achieving higher AUC-rates.

The analysis of current literature showed that multiple deep learning techniques are used for anomaly detection in various areas. In this research, the deep learning techniques autoencoder, variational autoencoder, long short-term memory neural network and spectral-residual convolutional neural network are proposed to detect anomalies in aquaculture data.

RESEARCH METHODOLOGY

This section presents the methodology for detecting anomalies in time series data for aquaculture. Initially, the data is analysed and pre-processed, and then deep learning algorithms are employed for anomaly detection. Specifically, four deep learning algorithms were employed, namely Autoencoder, Variational Autoencoder, Long-Short Term Memory neural network, and Spectral-Residual Convolutional Neural Network.

DATASET

The performance of anomaly detection was evaluated using real-life time-series datasets collected from three tanks in a salmon hatchery. The datasets were collected from pH, dissolved oxygen, and temperature sensors, which are commonly used in IoT aquaculture systems. These sensors were chosen due to their importance in maintaining the proper conditions for fish farming and preventing unwanted situations (Petkovski et al. 2021). The datasets consisted of nine sets of data collected between 1 June 2021 and 31 December 2021, with a per-minute frequency and a total of 308,160 records in each dataset.

DATA CLEANING

In this stage of the analysis, the missing values and duplicated timestamps were addressed. Any acquisitions that only had a timestamp and no corresponding sensor reading were identified as missing values and handled using spline interpolation techniques (Mathieu et al. 2017).

After handling all the missing values, it was observed that some timestamps in the datasets were not unique due to daylight-saving time. This change in time caused the clock to be pushed back one hour on 31 October 2021, resulting in duplicate keys in the time series data of the sensors. To fix this problem and make the time series usable, the two hours of acquisitions after 2:00 AM were condensed into a single hour, therefore removing the repeating timestamps. The two hours of data collected after 2:00 AM were merged into a single hour by computing the average value from two consecutive records to represent a single acquisition, effectively eliminating the duplicate timestamps.

AUTOENCODER

An autoencoder is a neural network architecture that has the input and output layers designed to be identical. A basic autoencoder consists of two parts: the encoder, which includes the front layers, and the decoder, which includes the rear layers. The middle layer constitutes the code, which serves as a latent representation of the input and is shared by both the encoder and the decoder. The middle layer of an autoencoder typically has significantly fewer neurons compared to the input layer. The code generated by an AE can be considered as the dimensionality reduction or feature extraction result of the input data. The AE takes the input data, transforms it into a code, and then reconstructs it from the code to produce the output. The difference between the input and the reconstructed input is known as the reconstruction error, which is minimized during the training process of the AE. (Li & Jiang 2020). The architecture of simple auto encoder is illustrated in Figure 1:

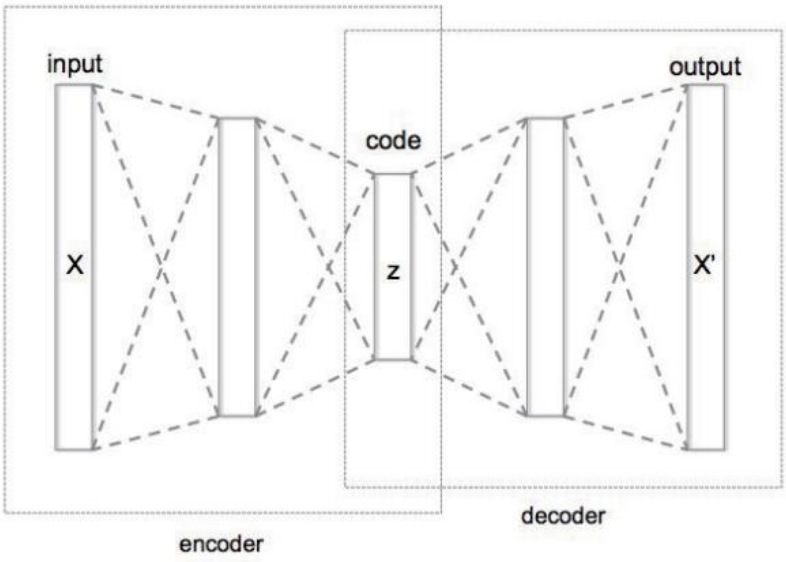


Figure 1 Architecture of Autoencoder

VARIATIONAL AUTOENCODER

The Variational Autoencoder (VAE) is a neural network consisting of an encoder and a decoder. The encoder network receives an input and produces a smaller representation, while the decoder network converts the smaller representation back to the original input. The encoder outputs two vectors, one for the means and one for the standard deviations/variances, for each latent attribute, forming the parameters of a vector of random variables from which the sampled encoding is obtained. While the mean and standard deviations are consistent for the same input, the actual encoding will differ because of sampling. The mean vector determines where the input should be centred, while the standard deviation controls how much the encoding can deviate from the mean. The VAE uses the Kullback-Leibler divergence to ensure smooth interpolation and construction of new samples in the loss function. The decoder generates a vector input by taking random samples from each latent state distribution and tries to recreate the original input. However, backpropagation cannot be used for random sampling, so the reparameterization technique is used instead. This technique optimizes the distribution parameter while maintaining the ability to randomly sample from that distribution. The challenge of the VAE is to learn a meaningful and generalizable latent space despite having far fewer units in the encoding than the input (Nugroho et al. 2020). The architecture of Variational Autoencoder is visualized on Figure 2:

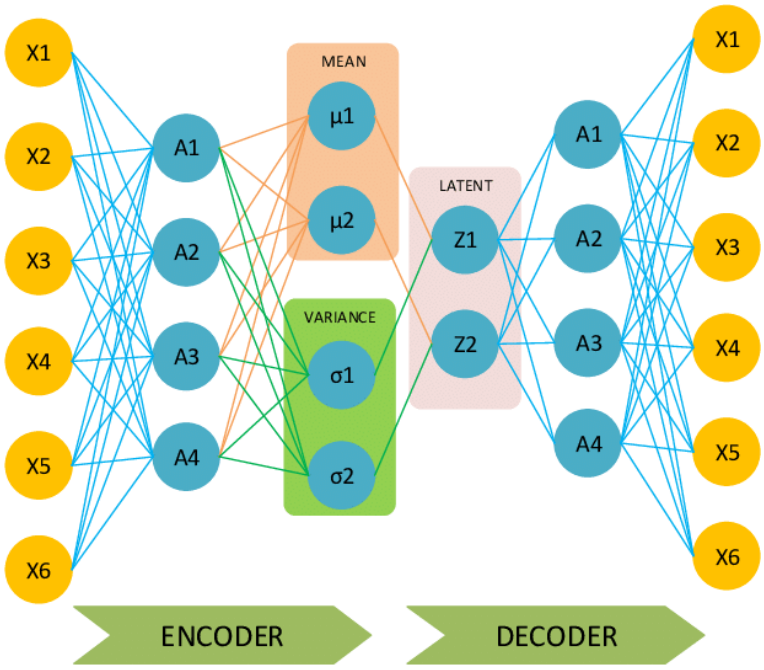


Figure 2: Variational Autoencoder architecture

LONG-SHORT TERM MEMORY AUTOENCODER

The Long Short-Term Memory (LSTM) neural network is a specialized type of Recurrent Neural Network that is capable of retaining memory over a long period of time. Unlike fully connected dense artificial neural networks that do not consider historical data, LSTM can overcome practical challenges such as vanishing gradients when using long sequences of input data. The LSTM architecture employs a modified backpropagation approach of gradient descent technique to overcome the vanishing gradient problem and enable the network to comprehend complex nonlinear patterns. The LSTM cells pass cell state to a network with minimal linear operations, and connecting these cells forms the architecture of an LSTM autoencoder.

An LSTM autoencoder is a type of artificial neural network that can learn how to efficiently encode data in an unsupervised manner. It consists of three main parts: an encoder, a latent vector, and a decoder. Throughout the training process, the encoder and decoder reduce input and reconstruct output via a compressed latent vector, aiming to keep network input as similar as possible to the network output. The primary distinguishing feature of an LSTM autoencoder compared to a typical autoencoder is that the primary building blocks of the network architecture are LSTM cells (Do et al. 2023). The architecture of LSTM-AE is illustrated in Figure 3:

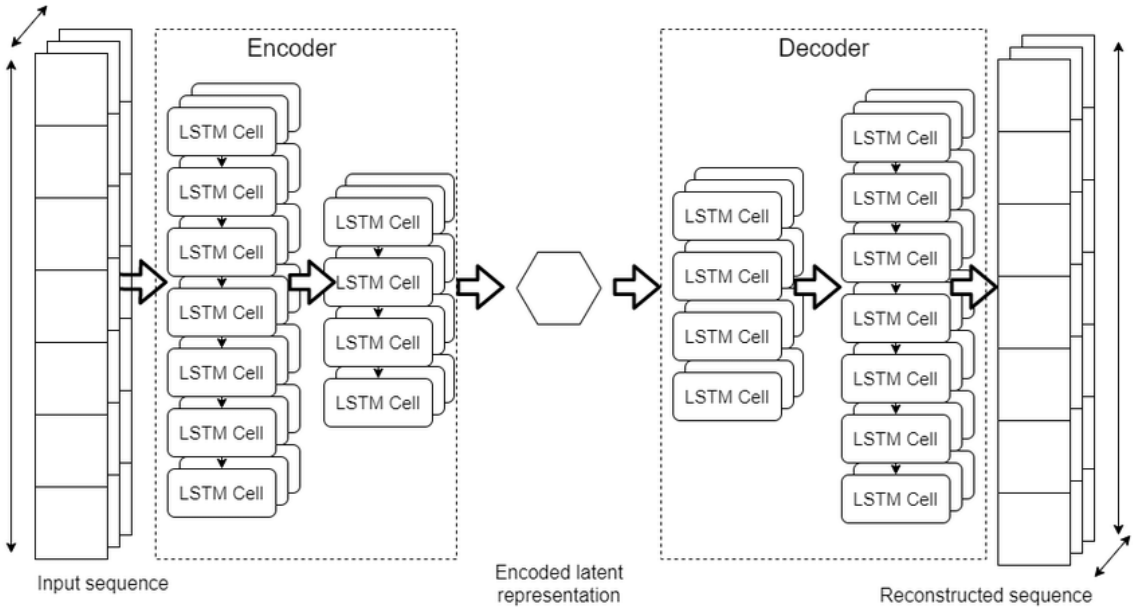


Figure 3: LSTM Autoencoder Architecture

SPECTRAL-RESIDUAL CONVOLUTIONAL NEURAL NETWORK

(Ren et al. 2019) proposed a novel algorithm that combines Spectral Residual (SR) and Convolutional Neural Network (CNN) for detecting anomalies in time-series data. The authors adapted the SR model, which is typically used for visual saliency detection, to perform unsupervised anomaly detection in time-series data. The SR algorithm involves three main steps: a Fourier Transform is applied to obtain the log amplitude spectrum, the spectral residual is calculated, and an Inverse Fourier Transform is applied to transform the sequence back into the spatial domain.

A Convolutional Neural Network (CNN) is chosen as the discriminative model architecture due to its commonly used supervised model for saliency detection. The CNN consists of two 1-D convolutional layers, where the channel size of the first layer is equal to the sliding window and the channel size is doubled in the second layer. Two fully connected layers are stacked before the Sigmoid output. The loss function used is cross-entropy, and the training process utilizes the SGD optimizer. The architecture of SR-CNN is visualized in Figure 4:

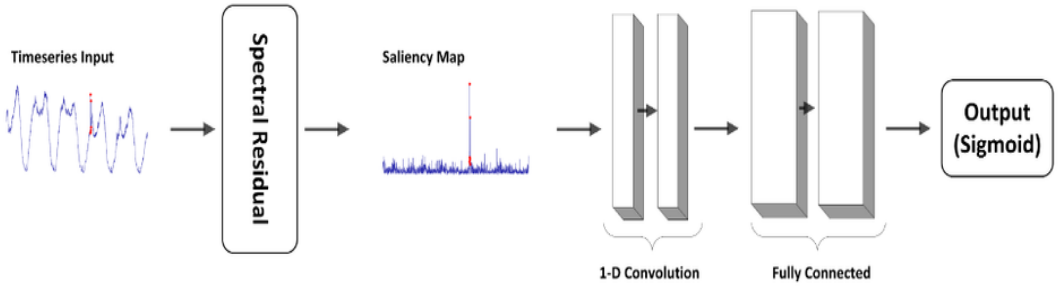


Figure 4: SR-CNN architecture

EVALUATION

The performance of all experimental methods is evaluated based on four selected evaluation metrics, namely accuracy, recall, precision, and F1-score (Demestichas et al. 2021). Precision is calculated by dividing the number of correctly detected anomaly points by the total number of detected anomaly points. On the other hand, Recall is computed by dividing the number of correctly detected anomaly points by the total number of actual anomaly points. Precision and Recall are two opposing measures, meaning that if one is high, the other is usually low. Specifically, high precision values are typically associated with low recall values, and vice versa. The formulae for calculating Accuracy, Precision, Recall and F-score can be seen below:

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

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

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

$$F - 1\ Score = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall}$$

The term True Positive (TP) is used to denote the number of correctly classified anomalies, while True Negative (TN) refers to the number of correctly classified normal events. The False Positive (FP) metric represents normal events that are mistakenly classified as anomalies, and the False Negative (FN) metric captures the anomalies that are wrongly identified as normal events (Cheng et al. 2019).

RESULTS

The results presented in this section test the performance of the deep learning algorithms for detecting anomalies in the sensors' datasets. The deep learning algorithms, AE,

VAE and LSTM-AE were implemented in the Python programming language on Jupiter Notebook IDE using the libraries tensorflow and keras (Braei & Wagner 2020). The SR-CNN algorithm was implemented using ML.NET, an open source and cross-platform machine learning framework provided by Microsoft (Mukherjee, 2021). The experiments were conducted on a PC with an AMD Ryzen 7 3700U processor (with 4 cores at 2.30 GHz) and 16.0 GB RAM.

Four criteria accuracy, precision, recall, and F1 score were considered to evaluate the models. The experiments were performed on nine datasets, three datasets per each sensor type. 40% of each time series was used as training set and the rest of the 60% data as test set. Unsupervised approach was used for training the models, and in the training process it was not used any label information. The data from the test sets were labelled and compared the anomaly detection results of each approach with the labelled data.

The obtained test results from the pH datasets showed that the SR-CNN algorithm outperformed the other algorithms. The SR-CNN algorithm attained the highest average accuracy of 84.6% and the highest average F1-score of 44.9% among all algorithms tested. The AE and VAE algorithms showed 100% precision, but the recall and F1-score were significantly low, indicating that these algorithms detected very few anomalies. The LSTM-AE algorithm outperformed the AE and VAE algorithms in accurately detecting the anomalies. The average accuracy rate achieved by the LSTM-AE algorithm was 81.5%, while its F1-score was 23.6%. However, the performance of LSTM-AE was still less effective than the SR-CNN algorithm in detecting anomalies. These results suggest that the SR-CNN algorithm is the most suitable algorithm for detecting anomalies in the pH dataset. The details from the results for the pH datasets are in the following table:

	Sensor	P	R	F1	ACC
AE	pH1	1.000	0.007	0.013	0.951
	pH2	1.000	0.001	0.002	0.745
	pH3	1.000	0.001	0.003	0.742
	Average	1.000	0.003	0.006	0.813
VAE	pH1	1.000	0.007	0.013	0.951
	pH2	1.000	0.001	0.003	0.745
	pH3	1.000	0.001	0.003	0.742
	Average	1.000	0.003	0.006	0.813
SR-CNN	pH1	0.145	0.308	0.198	0.885
	pH2	0.783	0.440	0.563	0.828
	pH3	0.741	0.485	0.586	0.825
	Average	0.556	0.411	0.449	0.846

LSTM-AE	pH1	0.544	1.000	0.704	0.959
	pH2	0.865	0.001	0.002	0.745
	pH3	0.114	0.001	0.002	0.740
	Average	0.508	0.334	0.236	0.815

The test results obtained from the oxygen datasets indicate that the LSTM-AE algorithm is the most effective in detecting anomalies. The LSTM-AE algorithm recorded the highest average accuracy of 97.4% and the highest average F1-score of 55.4%. The average precision of the LSTM-AE was 41.7%, and the average recall was 100%. In comparison, the AE and VAE algorithms achieved average accuracies of 92.7% and 91.9%, respectively, but their average precision and average recall were both below 20%. The SR-CNN was less effective with 86.2% average accuracy and 18.3% average F1 score. These results indicate that the LSTM-AE algorithm is the most suitable algorithm for detecting anomalies in the oxygen dataset. The details from the results for the oxygen datasets are in the following table:

	Sensor	P	R	F1	ACC
AE	Oxygen1	0.013	0.295	0.025	0.934
	Oxygen2	0.007	0.018	0.009	0.886
	Oxygen3	0.508	0.197	0.284	0.961
	Average	0.176	0.170	0.106	0.927
VAE	Oxygen1	0.012	0.295	0.024	0.928
	Oxygen2	0.005	0.018	0.008	0.868
	Oxygen3	0.512	0.197	0.284	0.961
	Average	0.176	0.170	0.105	0.919
SR-CNN	Oxygen1	0.014	0.545	0.027	0.881
	Oxygen2	0.131	0.703	0.221	0.849
	Oxygen3	0.187	0.785	0.302	0.857
	Average	0.111	0.678	0.183	0.862
LSTM-AE	Oxygen1	0.128	1.000	0.228	0.980
	Oxygen2	0.633	1.000	0.775	0.982
	Oxygen3	0.490	1.000	0.658	0.959
	Average	0.417	1.000	0.554	0.974

According to the test results obtained from the temperature datasets, the LSTM-AE algorithm outperformed the other algorithms in detecting anomalies. It achieved the highest average accuracy of 93.4% and the highest average F1-score of 51.8%. The average precision and recall for LSTM-AE were 85.3% and 48.2%, respectively. In contrast, the AE and VAE algorithms had average accuracies of 91.8% and 87.4%, respectively, with average F1 scores of 45.6% for AE and 34.1% for VAE. The SR-CNN algorithm also showed less effective results, with an average accuracy of 82.4% and an average F1-score of 39%. These findings suggest that the LSTM-AE algorithm is the most appropriate algorithm for detecting anomalies in the temperature dataset. The details for the performance of the models of temperature datasets are in the following table:

	Sensor	P	R	F1	ACC
AE	Temp1	0.656	0.280	0.393	0.951
	Temp2	1.000	0.485	0.653	0.935
	Temp3	0.319	0.326	0.322	0.868
	Average	0.658	0.364	0.456	0.918
VAE	Temp1	0.038	0.095	0.055	0.816
	Temp2	1.000	0.484	0.652	0.935
	Temp3	0.327	0.303	0.315	0.872
	Average	0.455	0.294	0.341	0.874
SR-CNN	Temp1	0.099	0.264	0.144	0.824
	Temp2	0.389	0.798	0.523	0.816
	Temp3	0.352	0.890	0.504	0.831
	Average	0.280	0.651	0.390	0.824
LSTM-AE	Temp1	1.000	0.115	0.206	0.950
	Temp2	1.000	0.523	0.687	0.940
	Temp3	0.560	0.809	0.662	0.920
	Average	0.853	0.482	0.518	0.934

CONCLUSION

In this paper, a comparative analysis and evaluation of deep learning techniques for anomaly detection in aquaculture is presented. Real-world datasets from pH, dissolved oxygen, and temperature sensors in three aquaculture tanks were used to evaluate the performance of four deep learning approaches: AE, VAE, LSTM-AE, and SR-CNN. The performance of the deep learning algorithms was tested on nine datasets, three datasets per each sensor type. Results revealed that the SR-CNN algorithm outperformed the other algorithms with an average accuracy of 84.6% and an average F1-score of 44.9% for detecting anomalies in the pH dataset. On the other hand, the LSTM-AE algorithm was found to be the most effective for detecting anomalies in the oxygen and temperature datasets, recording an average accuracy of 97.4% and 93.4%, along with an average F1-score of 55.4% and 51.8%, respectively. The AE and VAE algorithms were found to be less effective for detecting anomalies in all datasets.

The study was conducted on a large dataset of 308,160 records collected at a per-minute frequency between 1 June 2021 and 31 December 2021. The findings suggest that deep learning algorithms can effectively detect anomalies in aquaculture sensors datasets. In future research, more deep learning techniques will be explored along with new aquaculture datasets.

REFERENCES

1. Braei, M., & Wagner, S. (2020). Anomaly Detection in Univariate Time-series: A Survey on the State-of-the-Art. Computer Science - arXiv.
2. Choi, K., Yi, J., Park, C., & Yoon, S. (2021). Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines. IEEE Access, 9, 120043 - 120065.
3. Cook, A. A., Mısırlı, G., & Fan, Z. (2020). Anomaly Detection for IoT Time-Series Data: A Survey. IEEE Internet of Things Journal.
4. Demestichas, K., Alexakis, T., Peppes, N., & Adamopoulou, E. (2021). Comparative Analysis of Machine Learning-Based Approaches for Anomaly Detection in Vehicular Data. Vehicles, 3, 171-186.
5. Do, J. S., Kareem, A. B., & Hur, J.-W. (2023). LSTM-Autoencoder for Vibration Anomaly Detection in Vertical Carousel Storage and Retrieval System (VCSRS). Sensors.
6. Dupont, C., Cousin, P., & Dupont, S. (2018). IoT for Aquaculture 4.0 Smart and easy-to-deploy real-time water monitoring with IoT. Global Internet of Things Summit (GIoTS). Bilbao, Spain.
7. Encinas, C., Ruiz, E., Cortez, J., & Espinoza, A. (2017). Design and implementation of a distributed IoT system for the monitoring of water quality in aquaculture. Wireless Telecommunications Symposium (WTS). Chicago, IL, USA.
8. Gaddam, A., Wilkin, T., & Angelova, M. (2019). Anomaly detection models for detecting sensor faults and outliers in the iot-a survey. 13th International Conference on Sensing Technology (ICST). Sydney, Australia.
9. Hawkins, D. M. (1980). Identification of outliers (Vol. 11). Springer.
10. Li, Y.-L., & Jiang, J.-R. (2020). Anomaly Detection for Non-Stationary and Non-Periodic Univariate Time Series. 2nd IEEE Eurasia Conference on IOT, Communication and Engineering.
11. Liu, W., Jiang, H., Che, D., Chen, L., & Jiang1, Q. (2020). A Real-time Temperature Anomaly Detection Method for IoT Data. 5th International Conference on Internet of Things, Big Data and Securit.
12. Liu, Y., Pang, Z., Karlsson, M., & Gonga, S. (October 2020). Anomaly detection based on machine learning in IoT-based vertical plant wall for indoor climate control. Building and Environment, 183.
13. Mathieu, L., Aubin, J.-B., & Clemens, F. H. (2017). Interpolation in Time Series: An Introductive Overview of Existing Methods, Their Performance Criteria and Uncertainty Assessment. Water, 9(10).
14. Nugroho, H., Susanty, M., Irawan, A., Koyimatu, M., & Yunita, A. (2020). Fully Convolutional Variational Autoencoder for Feature Extraction. Journal of Computer Science and Information.
15. Petkovski, A., & Shehu, V. (2023). Anomaly Detection on Univariate Sensing Time Series Data for Smart Aquaculture Using K-Means, Isolation Forest, and Local Outlier Factor. Meco (12th Mediterranean Conference on Embedded Computing).
16. Petkovski, A., Ajdari, J., & Zenuni, X. (2021). IoT-based Solutions in Aquaculture: A Systematic Literature Review. 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO).

17. Ren, H., Xu, B., Wang, Y., Yi, C., Huang, C., & Kou, X. (July 2019). Time-Series Anomaly Detection Service at Microsoft. KDD '19: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, (pp. 3009–3017).
18. Xu, H., Feng, Y., Chen, J., Wang, Z., Qiao, H., Chen, W., . . . Pei, D. (2018). Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications. World Wide Web Conference.
19. Cheng, Z., Zou, C., & Dong, J. (2019). Outlier detection using isolation forest and local outlier factor. RACS '19: Proceedings of the Conference on Research in Adaptive and Convergent Systems, (pp. 161–168). Gdansk, Poland.
20. Mukherjee, S. (2021). Anomaly Detection. In: ML.NET Revealed. Apress, Berkeley, CA.