IRAQI STATISTICIANS JOURNAL

https://isj.edu.iq/index.php/isj

ISSN: 3007-1658 (Online)

Proposing a Quantitative CNNs Model for Multivariate Prediction

Bashar Khalid Ali[1], Adel Abbood Najm[2] and Mahdi Wahab Nea'ama [3]

[1] Department of Statistics, Administration and Economics College, Kerbala University-Iraq
[2] Department of Statistics, Administration and Economics College, Sumer University-Iraq
[3] AlSafwa University College- Kerbala- Iraq

ARTICLE INFO

Article history:

Received	14/11/2024
Revised	15/11/2024
Accepted	14/1/2025
Available online	15/5/2025

Keywords:
Convolutional Neural Network
Shark Smell Algorithm
Optimization
Natural Gas Prices
Prediction

ABSTRACT

Convolution Neural Network (CNNs) are one of the effective approaches in image processing, pattern recognition, and video analysis, but there is a gap in research related to their application to quantitative prediction of multiple variables. In this article, a new algorithm called Quantitative CNNs is proposed, which is used to predict quantitative data by modifying the network structure instead of being prepared to process images. It can be adapted to work with temporal data or multidimensional quantitative data by processing convolution layers after passing the data through the convolution layers, and using dense layers. During the training step, we use quantitative data as inputs and provide the numerical results that need to be predicted as target values, so the network outputs are quantitative numbers instead of classifications by enhancing the training process employing the shark smell optimization (SSO) method. The method was also applied to a real multi-data set representing natural gas prices, namely spot price, futures price, storage level, daily demand, temperature, shipping and transportation costs, geopolitical events, oil prices, and currency exchange rate (Iraqi Dinar vs. US Dollar). We conclude that the proposed algorithm. The proposed algorithm was effective in estimating quantitative time series data compared to the traditional neural network model, and that the shark smell algorithm which used to improve the prediction accuracy gave more accuracy to the method according to the criteria RMSE and Accuracy. The analysis process was carried out using MatLab version 2023b.

1. Introduction

Quantitative time series are a complex type of data that require advanced analytical tools to understand their patterns and future trends. These series are used in many vital applications, such as price prediction or forecasting in financial markets, climate data analysis, industrial system monitoring, and energy demand prediction or forecasting. Although there are many traditional statistical models for analyzing these series, such as autoregressive models and moving averages (Autoregressive integrated moving average(ARIMA)), these models face significant challenges in dealing with nonlinear patterns and complex interactions between data.

With the great development in the field of artificial intelligence, neural networks have emerged as effective tools for analyzing multidimensional and complex data. Among these tools, (CNNs) have proven effective in image processing, as they extract basic features such as edges, shapes, and color gradients. These networks rely on applying small filters (kernels) to the data to extract important patterns. Many researchers have studied convolutional neural networks as an image analysis technique. (Wu et al., 2022) used convolutional neural networks to predict the

[*] Corresponding author. E-**mail** address: bashar.k@uokerbala.edu.iq

https://doi.org/10.62933/g5mk7247

remaining time of initially loaded gravel and the time period of coupled flow images of gravel bed of nuclear reactors [1]. (Dumakude & E. Ezugwu, 2023) performed automated detection of COVID-19 using convolutional neural networks [2]. (Zhao et al., 2024) used convolutional neural networks in image classification, object detection, and video prediction [3].

In this paper, we extend the scope of application of convolutional neural networks to suit the analysis of quantitative time series. The main goal is to exploit the ability of these networks to extract essential features from temporal data, such as frequency, seasonal fluctuations, and long-term trends. To achieve this, the architecture of convolutional neural networks is adapted to include convolution layers, nonlinear activation, and pooling techniques, allowing for efficient analysis of temporal patterns. Furthermore, to improve the prediction accuracy, the shark smell algorithm, an optimization algorithm inspired by the biological behavior of sharks in search of food, is incorporated. This algorithm contributes to improving the performance of the model by adjusting the network weights in a way that increases the prediction accuracy and reduces the errors.

Through this methodology, this article aims to develop an innovative model that can accurately and efficiently predict quantitative time series, while providing an advanced analytical tool that helps to explore and comprehensively analyze complex patterns in temporal data.

The proposed method gains its importance from its ability to handle complex and nonlinear quantitative time series data that traditional models such as ARIMA cannot handle. It is based on convolutional neural networks (CNNs), which have proven their effectiveness in image and video analysis, and extends their use to suit time series analysis. These networks are characterized by their ability to extract essential features from data, such as frequencies, seasonal fluctuations, and long-term trends, making them a powerful tool for analysing temporal data. To improve the prediction accuracy and reduce errors, the shark smell algorithm, inspired by the biological behaviour of foraging, is incorporated and used to optimally adjust the network weights and improve its performance. This methodology presents an innovative model that combines deep analysis of time series and prediction accuracy, opening new horizons for practical applications in areas such as price prediction in financial markets, climate data analysis, industrial system monitoring, and energy demand prediction or forecasting.

2. Convolution neural networks:

It is an algorithm that uses deep learning introduced by the French computer scientist (Fukushima, 1980) [4], and developed by (LeCun et al., 1998) [5], and is also known as a filtration neural network. It is used to recognize objects, such as classifying images. Much like how brain cells function in a living organism [6]. It has a major role in many fields; including computer vision aims to teach computers to understand visual data as human vision, as well as facial recognition, medical image analysis, and many other real-world applications such as surveillance camera systems and self-driving cars. [7][8] It is named after the mathematical operation called convolution that was used in its construction and operation, by which a filter or kernel is passed over any part of the input image and the filter is moved in different parts of the image to recognize its characteristics and thus obtain a feature map that includes important extracted information such as edges, corners or patterns specific to the image, making it a powerful tool for image recognition and feature discovery [9]. However, in traditional neural networks, its work is like a tree, since an input layer is a part of it, a hidden layer or layers, output layer, where observations to be in one direction only, from input to the output layers, and no loops or feedback between the layers with dense layers to lessen the data set's overall size and obtain more predictive ability, it also includes pooling layers [10][11].

3. CNNs Structure

Applying a mathematical convolution operation to the input components representing neurons from the previous layer or pixels from the input image such that each set of input components undergoes a convolution operation of a single filter size, resulting in feature maps with a single value. We then create a new feature map by repeatedly performing all the operations mentioned above for each filter. The weights of the network are the values of these filters. Here is an analysis of these layers: [12] [13]

3.1 Shared Weights

It is the process that includes repeating the filter convolution process on all input elements by providing a large number of connections but with shared weights between them, which leads to recognizing the image characteristics based on those connections based on their original location by linking neighbouring points to each other and thus allowing the representation of overlapping regions, so that learning becomes more efficient and makes the network qualified to achieve better results [4].

3.2 Duplicate filters

The process of using multiple filters with the same inputs and applying the convergence process to them will contribute to identifying different sets of features in the image, each of which will give us unique characteristics, considering that the number of filters placed such that it does not cause the complexity of the computational operations by considering that the number of examples available for training, the complexity of the process, and the variance and dimensions of the input image, while dimensions a single filter depend on the data in the set of training examples [4] .

3.3 Sub-sampling layers

If considering that reduce neurons number, we put sub-layers after each layer of the convolutional layers to shorten each group of input neurons to one neuron with an ideal value of 2x2 in several ways, including taking the highest value among them, taking mean values, or taking fully connected layers [4]. These layers connect all the neurons for previous layer and make them inputs for each neuron, as in normal neural networks, and a specific number must be considering, but there are often two consecutive layers of them as the last layers of the network, as it cannot come before a layer of the convolutional type [14].

4. Mathematical Model for CNNs:

Convolutional neural networks (CNNs) use an advanced mathematical structure to recognize patterns and extract features from data with a network structure, such as images. The mathematical model for these networks is based on a set of basic mathematical operations, namely convolution, activation, pooling, and fully connected as follows:

4.1 Convolution:

The convolution operation is a basic operation in convolutional neural networks and is done using filters (kernels) that pass over the image (or data) to extract features. The mathematical equation for convolution is: [8][15]

Convolution is a fundamental operation in convolutional neural networks and is done using filters (kernels) that are passed over the image (or data) to extract features. The mathematical equation for convolution is: [8][15]

$$O(i,j) = \sum_{u=0}^{m-1}\sum_{v=0}^{n-1} k(u,v)\, I(i+u, j+v) \qquad (1)$$

Where $O(i,j)$ the output value at position (i,j) after applying the convolution operation. In other words, it's the value in the resulting "feature map" at a specific position (i,j) .This value is calculated by applying the filter (kernel) to a specific region for input image centered around(i,j). The map O is essentially a transformed version of the input image, highlighting certain features (like edges or textures) based on the applied filter. Each position (i,j) in O holds a numerical value that represents the presence or intensity of a specific feature at that location in the input image, I is the input matrix (image), $k(u,v)$ the kernel value at position (u,v) ,$I(i+u,j+v)$ are convolved the pixel value of the image.

4.2 Activation Layer:

Adding nonlinearity to the model by applying the activation function post-convolution.

Convolutional neural networks make use of a set of standard activation functions as follows, : [16]

1. **Sigmoid:** often used in convolutional neural networks. Due to its usefulness for binary classification tasks, its range is between 0 and 1as follows:[16]

$$\sigma_x = \frac{1}{1+e^{-x}} \qquad (2)$$

This function keeps positive values as they are and zeroes out negative values.

2. **Hyperbolic Tangent (Tanh)**: stands for hyperbolic tangent, is another popular tool for RNNs. Its usefulness for nonlinear classification problems is enhanced by its range of 1 to -1as follows:[16]

$$\sigma_x = \frac{e^x - e^{-x}}{e^x + e^{-x}} \qquad (3)$$

3. **Rectified Linear Unit (ReLu**.): Often used in deep neural networks; it is non-linear. For models that need positive outputs, its range of 0 to infinity is ideal as follows:[17]

$$\sigma_x = \max(0, x) \qquad (4)$$

Its usage and prevalence are unparalleled in convolutional neural networks (CNNs), rather than Sigmoid or Tanh which are more common in RNNs, as ReLu helps convolutional networks perform better by effectively introducing nonlinearity while reducing complex computations.

4. **Leaky ReLu**: In the same vein as ReLu, but with the addition of a tiny slope to negative values to shield the model from "dead neurons" as follows:[16]

$$\sigma_x = \max(0.01x, x) \qquad (5)$$

5. **Softmax**: used in convolutional neural networks' output layers for multi-class classification jobs. A probability distribution across all potential classes is generated from the network output as follows:[16]

$$\sigma_x = \frac{e^x}{\sum e^x} \qquad (6)$$

4.3 Pooling Layer:

The dimensions of the matrix reduced and generated using the convolution layer, which reduces computational complexity and helps avoid over-fitting. There are two basic types of pooling:[18]

4.3.1 Max Pooling: reduce the dimensions of matrix generated using the convolution layer, while preserving the most important features. In max pooling, a window (usually 2x2 or 3x3) is selected that passes through the image from the convolution layer. For each window location, the highest value (maximum value) is extracted from that window to become the value represented for that location in the resulting feature map: [18]

$$\boldsymbol{P(i,j)} = \mathbf{Max}_{\boldsymbol{R \in (n,m)}}\, \boldsymbol{I}\,\boldsymbol{(i,j)} \qquad (7)$$

Where $\boldsymbol{I(i,j)}$ values inside the collection window, R the receptive field or window which used to apply maximal pooling.

4.3.2 Average Pooling:

It reduces the dimensionality that generated from the convolution layer, but in a different way than max pooling, a window (usually 2x2 or 3x3) is passed through it extracted from the convolution layer. For each window location, the values in that window are averaged, and this value becomes the representation of that location in the resulting feature map: [18]

$$\boldsymbol{P(i,j)} = \boldsymbol{I(i,j)} \sum_{\boldsymbol{R \in (n,m)}} \frac{\mathbf{1}}{|\mathbf{R}|} \qquad (8)$$

4.3.3 Fully Connected Layers:

They are used in the end to produce the final prediction. Each node connected in previous layer. They play a key role in aggregating the features extracted from the previous layers and generating the final output, whether it is classifications or predictions with completely linked layers, every node may access and use all the features from the layers below it. This allows for comprehensive feature analysis at all levels, as shown by the following mathematical equation:[19]

$$\boldsymbol{y = Wx + by} \qquad (9)$$

Where Y is the output, W the matrix of weights, X input matrix, b bias vector.

5. General application of the mathematical model:

A convolutional neural network consists of several successive layers of convolution, activation, and pooling, followed by fully connected layers as follows:[20]

$$\boldsymbol{F(x) = fL\big(WL \cdot \big(\ldots f2\big(W2 \cdot \big(f1(W1 \cdot x + b1)\big) + b2\big) \ldots\big) + bL\big)} \qquad (10)$$

Where x initial input, W, b weights and biases of each layer, f activation function, L the number of layers in the network. Each layer conducted on a set of weights and biases to input, and then the result passed through the activation function f. The last layer, L, is where the model's output, F(x), is generated by repeating this procedure over all levels.

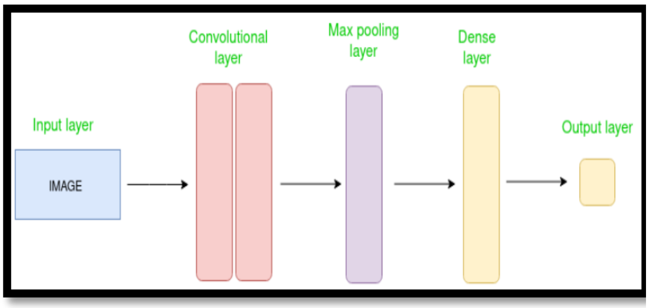

Figure 1. CNNs Architecture

Figure 1 showed that several filters were applied to analyse and extract specific features from a random colour image. The first filter is the horizontal edge detection filter, which uses a kernel matrix that calculates the difference between pixel values in the vertical direction, highlighting only horizontal edges. The second filter is the vertical edge detection filter, which applies a matrix that detects vertical edges by calculating the difference between pixel values in the horizontal direction. Next, a Gaussian Blur filter was applied, which uses a Gaussian smoothing matrix, which reduces image detail and removes noise by giving higher weight to pixels closer to the canter. The fourth filter is the Gradient Magnitude filter, which captures sharp changes in colour by calculating the gradient in directions x , y, highlighting large colour changes. 45° diagonal edge detection filter and the 135° diagonal edge detection filter use dedicated matrices to detect diagonal lines in both directions, which help in detecting different diagonal details. Finally, the contrast enhancement filter increases the clarity of details by highlighting colour differences. After applying these filters, the ReLU activation output of the colour neural network shows the extracted important features, where only positive values are retained, which emphasizes the salient parts of the image, and shows how the neural network simplifies and extracts the essential features.

6. Shark Smell Optimization (SSO)

Every living thing has an olfactory system that acts as its main receptor for distant chemical signals. The olfactory pits, found on the sides of fishes' heads, are where the smell receptors are situated. The water enters and exits the pit via two outside holes. Small hairs on the pit lining cells and the force from moving fish in water provide the mechanism that moves water within the pit. The olfactory nerve terminals have a pleated surface that dissolved substances may attach to [21]. The olfactory receptors in vertebrates are unique among sense nerves in that they link directly to the brain, bypassing any intermediate nerves.

A region of the cerebral cortex called the olfactory bulb receives signals related to scent. An olfactory pit houses both of a fish's olfactory bulbs. Fish have heightened sense of smell due to the fact that their olfactory pits are bigger and more specifically devoted to smell nerves, and because their brains have larger smell information centres [22]. The biggest olfactory bulbs, which are used for processing scent information, are found in eels and sharks. A little over 400 million years ago, nature's top predators—the sharks—made their debut in the water. One of the ways sharks stay alive in nature is by using their acute sense of smell to catch food. A shark's acute sense of smell is one of its strongest senses, sharks breathe via their nostrils, which are situated on each side of their snout, while they swim. Once the water reaches the olfactory pits, it travels via the sensory cell-lined skin folds. Because of their highly developed sensory cells, some sharks can smell even the faintest hint of blood [23]. A single drop of blood in a big pool, for instance, may be detected by a shark. Thus, a wounded fish may be detected by being able to detect of smell up to distant of kilometre [24].

Being able to detect of smell might serve as a navigational aid. As it travels from shark's left side into right pit, the smell goes beyond the left pit. Sharks use this method to pinpoint where an unpleasant smell is coming from. Figure 2 is a schematic depicting the shark's trajectory as it approaches the smell's origin. The ability to focus is crucial for the shark to make this manoeuvre and reach its target. To rephrase, the shark's actual movement is dictated by the attention level. An optimization algorithm may be built to discover the best solution to a problem based on this attribute as figure 3.

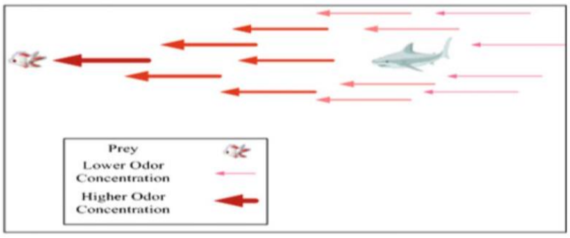

Figure 2. Diagram showing the shark's path to the smell's origin [24]

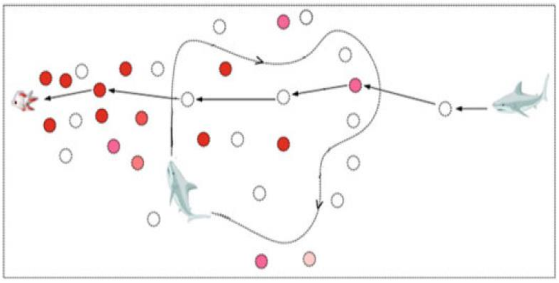

Figure 3. Rotational movement of a shark [24]

6.1 SSO Formulation

In order to build the model, some assumptions are taken into account, they consist of:

1. When fish sustain injuries, they bleed to death, which then ends up in the ocean (the search space). Consequently, the damaged fish's speed is irrelevant when compared to the shark's moving speed. To rephrase, it is believed that the source, which is the prey, remains constant.
2. An infusion of blood into salt water happens often. There is a disregard for how the passage of water affects smell-distorting particles. The smell particles are more concentrated in the area around the wounded fish, as is clearly seen. A shark may more easily approach its prey if it follows the olfactory particles.
3. In a shark's search habitat, a single wounded fish produces a smell.

6.2 SSO Initialization: Locating First Smell Particles

As soon as the shark detects an smell, it starts to hunt. Actually, smellous particles diffuse weakly from a wounded fish (prey). For the purpose of modeling this process, begin in start solutions to an optimization problem in the viable search space are produced at random. At the outset of the search, each of these answers stands in for an smell particle that might indicate the shark's location: [21][25]

$$\mathbb{X}_1^1, \mathbb{X}_2^1, , \mathbb{X}_3^1, \dots , \mathbb{X}_{NP}^1 \qquad (11)$$

(X_i^1) the i^th initial solution or location in the population vector, NP is the population size.

One way to represent the linked optimization issue is as:[21]

$$\mathbb{X}_i^1 = \mathbb{X}_{i,1}^1, \mathbb{X}_{i,2}^1,\ \mathbb{X}_{i,3}^1, \dots ,\ \mathbb{X}_{i\,,Nd}^1\ ;\ i=1, 2, \dots, ND, \qquad (12)$$

One way to represent the amount of choice variables in an optimization problem is by ND, and $\mathrm{X}_{\mathrm{i,j}}^1$ is either the j^th dimension of the location shark's i^th :[21][25]

6.3 Shark Approach to Prey:

The shark makes a beeline for its prey in every case. One way to express the initial velocity vector is by utilizing the position vectors as:[16][28]

$\mathbb{V}_1^1, \mathbb{V}_2^1,, \mathbb{V}_3^1, \dots, \mathbb{V}_{NP}^1$ (13)

The vectors of speed in Eq. (10) incorporate elements across all dimensions.

$\mathbb{V}_i^1 = \mathbb{V}_{i,1}^1, \mathbb{V}_{i,2}^1,\ \mathbb{V}_{i,3}^1, \dots,\ \mathbb{V}_{i\,,ND}^1$; i=1, 2, …, ND, (14)

A shark's course of action is dictated by the strength of a smell, which it follows. As the concentration of smell increases, the shark's velocity increases as well. From an optimization standpoint, the objective function's gradient quantitatively represents this change. The gradient shows the direction of the function's maximum rate of growth. The following equation shows this procedure, [21][25]

$\mathbb{V}_i^K = n_K R1.\nabla\,(\mathrm{OF})|_{\mathbb{X}_i^K}$; i=1, 2, …, NP, K=1,2,.., K_{Max} (15)

$\mathbb{V}_i^K$ The objective function, OF, represents the almost constant shark velocity, and ∇ is its gradien, k_{max} is The maximum number of phases for the shark's forward movement is denoted by k, n_K is a value in the interval [0,1], and R_1 is a randomly distributed value in the interval [0,1].[21][25]

Since the gradient function predicts a speed that a shark is incapable of achieving, n_K is in the interval [0,1]. The option R_1 is used to increase the SSO algorithm's inherent random search. At first, the idea of considering R_1 came from the gravitational search algorithm (GSA). Using is one approach to finding the velocities in all three dimensions. You may calculate the velocity in each dimension using this equation:[21][25]

$\mathbb{V}_{i.j}^K = n_K R1.\frac{\partial(OF)}{\partial \mathbb{X}_j}|_{\mathbb{X}_{i.j}^K}$; i=1, 2, …, NP, j=1,2,…, ND , K=1,2,.., K_{Max} (16)

The presence of inertia means that the shark can only accelerate so far; its current speed is directly proportional to its starting velocity. This process may be represented using a modified version of the equation (16) as follows:[21][25]

$\mathbb{V}_{i.j}^K = n_K R1.\frac{\partial(OF)}{\partial \mathbb{X}_j}|_{\mathbb{X}_{i.j}^K} + \alpha_j R2\mathbb{V}_{i.j}^{K-1}$; i=1, 2, …, NP, j=1,2,…, ND , K=1,2,.., K_{Max} (17)

The rate of momentum, α_j, is an inertia coefficient or rate of momentum that ranges from 0 to 1 at stage k. The momentum term is designed to be using random number generator with a uniform distribution over the interval [0, 1]. A higher value of k indicates a larger inertia and a stronger relationship between the current and previous velocities. Utilizing momentum in mathematics facilitates a more seamless traversal of the space of potential solutions throughout the search process. The search diversity of the algorithm is improved by R_2. You may ignore or provide the initial shark velocity $\left(\mathbb{V}_{i.j}^1\right)$ before the search process begins or assign it a very modest value while calculating first stage ($\mathbb{V}_{i.j}^0$).

Up to a certain point, it is possible to increase the shark's speed. The swim bladders that assist most fishes remain afloat are not present in sharks. Therefore, even at a slow pace, they can't stand still and must swim upwards. This process is propelled by the strong tail fin. While most sharks travel at a pace of about 20 km/h, aggressive sharks may reach speeds of 80 km/h or more. For instance, you can't have more than 80/20 or 1/4 as ratio of sharks' top to bottom velocities. Every step to SSO algorithm makes use of the following velocity limiter:[21][25]

$|\mathbb{V}_{i.j}^K| = \left[\left|n_K R1.\frac{\partial(OF)}{\partial \mathbb{X}_j}|_{\mathbb{X}_{i.j}^K} + \alpha_j R2\mathbb{V}_{i.j}^{K-1}\right|, \left|\beta_j \mathbb{V}_{i.j}^{K-1}\right|\right]$(18)

Given that stage k's velocity limiter ratio is represented by β_j. The sign $\mathbb{V}_{i.j}^K$, which is determined by Eq. (18), is identical to that chosen minimal operator in same equation. Because the shark is moving forward, its new location, denoted Z by $\mathbb{Y}_i^{K+1}$, is calculated by taking its prior position and velocity into consideration, [21][25]

$\mathbb{Y}_i^{K+1} = \mathbb{X}_i^K + \mathbb{V}_i^K \varDelta.t_k$; i=1, 2, …, NP, K=1,2,.., K_{Max} (19)

As the time interval of stage k is denoted by $\varDelta.t_k$. Let us assume, for the sake of argumentation, that $\varDelta t_k$ is one for all phases. $\left(\mathbb{V}_{i,j}^K, \mathrm{j} = 1,2,\dots,\mathrm{ND}\right)$ used to derive each component of vector $\mathbb{V}_i^K$ calculated from equation (19). [21][25]

$\mathbb{Z}_i^{K+1,m} = \mathbb{Y}_i^{K+1} + R3.\mathbb{Y}_i^{K+1}$ (20)

Where $\mathbb{Z}_i^{K+1,m}$ is equal to 1, m index of the point at local search begins, and R_3 a uniformly distributed random integer.

Sharks may detect stronger smell particles by both swimming ahead and rotating in their direction of travel. Really, according to Yao-Tsu (1971), this really helps it advance. Figure 4 shows the simulated version of this motion. The shark's rotatory motion follows a closed.

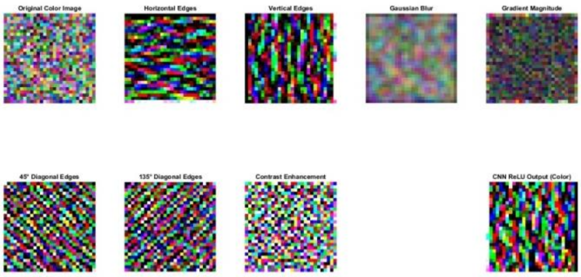

Figure 4. Input Image after applied seventh filters and output ReLu activation function

The range [−1, 1] may use to assess the limit of R$_3$ given that operator executes a local search around $\mathbb{Y}_i^{K+1}$. In the local search, M points are located near $\mathbb{Y}_i^{K+1}$ if the random number generator returns a value of zero. Just like a shark's circular movement, the connections of M points gain a closed shape. While swimming about, a shark will start its hunt from a spot with a stronger scent. One way to describe this feature of the SSO algorithm is as, [25]

$$\mathbb{X}_i^{K+1} = \operatorname{argmax}\left\{\mathrm{OF}(\mathbb{Y}_i^{K+1}), \mathrm{OF}\left(\mathbb{Z}_i^{K+1,i}\right) \ldots., \mathrm{OF}\left(\mathbb{Z}_i^{K+1,M}\right)\right\} \ ; \ \mathrm{i} = 1, 2, \ldots, \mathrm{NP} \qquad (21)$$

7. Proposed CNNs

Traditional CNNs deal with images by processing them through main layers called convolutional layers, the aim of which is to extract features in the image through either its edges, shape, colour gradient, or dimensions, etc. During the training process, these layers pass a filter on the image, which is a kernel and is represented by a small matrix of weights with a size of (3x3) or (5x5) by multiplying each value in the kernel by its corresponding value in part of the inputs, then the results are collected to give a new value. By passing these filters step by step and moving by moving the filter by one unit each time, by moving it by two units and adding a frame of zeros around the inputs so that the feature map maintains the same basic size of the inputs, this results in a map called the feature hole, which is the features extracted from the input image.

In order to transform convolutional neural networks to fit quantitative time series data, we will adapt the structure of these networks, to analyse time series or multidimensional data, instead of images. Suppose we have a quantitative time series $\mathrm{x}_1, \mathrm{x}_1, \ldots, \mathrm{x}_t$, t is the length of the time series, $\mathrm{x}_t \in R^n$ and let us also assume that the weights $\mathrm{w}_1, \mathrm{w}, \ldots, \mathrm{w}_k$, k represents the size of the kernel (core), then the convolution process between X and W at a certain point in time t can be defined as follows:[26]

$$h_t = \sum_{i=1}^{k} w x_{t+i-1} \qquad (22)$$

Where h represents the feature map extracted from the time series at location t resulting from moving the filter through the time series step by step through a (stride) stage, which helps to capture the patterns present in the time series data such as highs, lows, seasonality, fluctuations, etc. After obtaining the output of the striking kernel h_t, it is passed through a nonlinear activation function to deal with complex patterns: [26]

$$a_t = \max(0,\ h_t) \qquad (23)$$

Since a_t is the result of the activation function, which helps the network to capture nonlinear relationships in the data. Then the feature map is reduced in size using the pooling layer as follows:

$$t_p = \max(a_t, a_{t+1}, \ldots., a_{t+m-1}) \qquad (24)$$

Where m is the pooling size, t_p is the result of pooling across location t, which allows the selection of the most important features at each location of the time series. The extracted features are then passed through many dense layers to pool the patterns and prepare the final prediction as follows:

$$y = f(w.z + b) \qquad (25)$$

Where z input to the final dense (fully connected) layer in the network. Usually the flattened or aggregated feature map generated by the preceding convolutional and pooling layers. Essentially, it contains the key features learned by the network from the time series data, which are then used to make the final prediction in the last layer as follows,

$$z = f_{pool}\left(f_{activation}\left(f_{conv}(x)\right)\right) \qquad (26)$$

Where $\mathrm{f}_{\mathrm{conv}}$ the convolution function that applies filters over the time series. $\mathrm{f}_{\mathrm{activation}}$ Activation function applied after convolution (such as ReLu). $\mathrm{f}_{\mathrm{pool}}$ Pooling function that reduces the feature dimensions. w the weight matrix, b is the bias, and f is the final activation function which is always linear. Then we use the shark smell algorithm to improve the prediction accuracy.

8. Real data set:

Data on world natural gas downloaded from trading economics site from 1/1/2024 to 1/10/2024 were used, its Spot price of gas in US dollars, represent the direct price of natural gas in the market, is measured based on the amount of energy contained in the gas, which is measured in million British thermal units (MMBtu). Futures price its price that is determined for natural gas in the future based on the contracts concluded, and depends on the amount of energy in the gas in US dollars per million British thermal units (MMBtu). Storage level in billion cubic feet its amount of natural gas stored is measured in billion cubic feet (Bcf). Daily demand, the amount of gas consumed daily is measured in billion cubic feet in billion cubic feet (Bcf). Temperature its ambient temperature that may affect the demand for natural gas, as a decrease or increase in temperature affects energy use and in degrees Celsius (°C). Shipping costs, the unit of measure, its cost required to transport natural gas is measured based on the amount of energy contained in US dollars per million British thermal units ($/MMBtu). Geopolitical events, its geopolitical situation is measured qualitatively; such as "stable" or "tense", and reflects factors that may affect supply and demand, and are not quantitative and are classifications. Oil price its market price of crude oil per barrel, which can be linked to natural gas prices in US dollars per barrel ($/barrel). Exchange rate, the exchange rate shows the value of the Iraqi dinar compared to the US dollar as table (1)

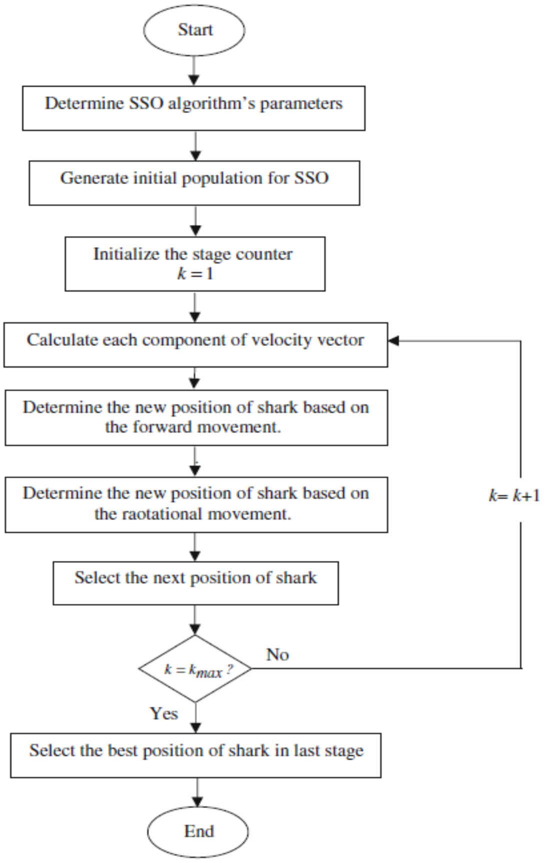
Start
Determine SSO algorithm's parameters
Generate initial population for SSO
Initialize the stage counter
k = 1
Calculate each component of velocity vector
Determine the new position of shark based on the forward movement.
Determine the new position of shark based on the raotational movement.
k= k+1
Select the next position of shark
k = kmax ?
No
Yes
Select the best position of shark in last stage
End

Figure 5. Flowchart of SSO Algorithm (Mohammad-Azari et al., 2017, 99)

Table 1. Real data set [27]

Date	Spot price of gas	Futures price	Storage level	Daily demand	Temperature	Shipping costs	Geopolitical events	Oil price	Exchange rate
01/01/2024	3.5	3.6	3500	120	15	0.5	stable	70	1480
01/02/2024	3.4	3.5	3480	122	14	0.52	stable	71	1481
01/03/2024	3.6	3.7	3470	125	13	0.51	tense	72	1482
01/04/2024	3.7	3.8	3460	127	12	0.53	stable	73	1480
01/05/2024	3.8	3.9	3450	130	11	0.54	tense	74	1479
01/06/2024	3.9	4	3440	128	10	0.55	stable	75	1480
01/07/2024	4	4.1	3430	130	9	0.56	stable	76	1481
01/08/2024	3.95	4.05	3420	129	8	0.57	tense	77	1482
01/09/2024	4.1	4.2	3410	132	7	0.58	stable	78	1483
01/10/2024	4.15	4.25	3400	134	6	0.59	stable	79	1482

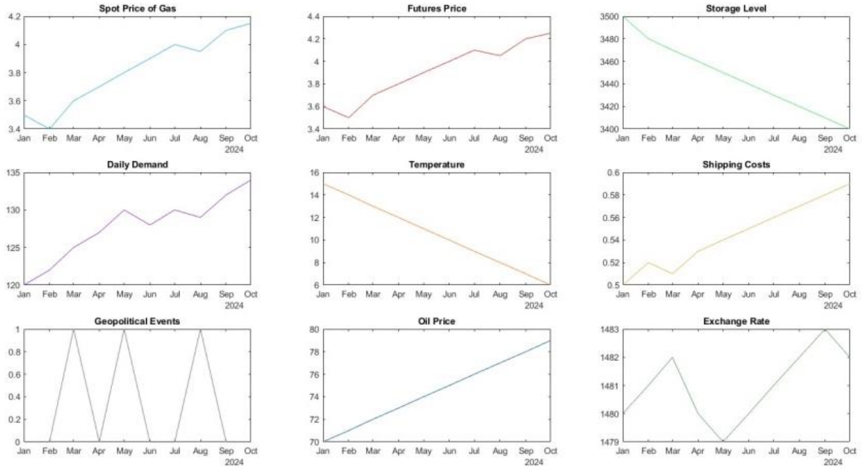

Figure 6. Curves of natural Gags variables

Table 2: Actual and Predicted Spot price of gas by NNs and QCNNs without optimization algorithm

windows size	Actual	NNs	QCNNs	RMSE		Accuracy	
				NNs	QCNNs	NNs	QCNNs
1	3.80	3.62	3.76	0.10	0.13	0.94	0.95
2	3.90	3.71	3.85	0.16	0.09	0.95	0.97
3	4.00	3.80	3.94	0.24	0.14	0.94	0.97
4	3.95	3.85	3.90	0.20	0.10	0.95	0.97
5	4.10	3.89	4.03	0.34	0.22	0.91	0.94

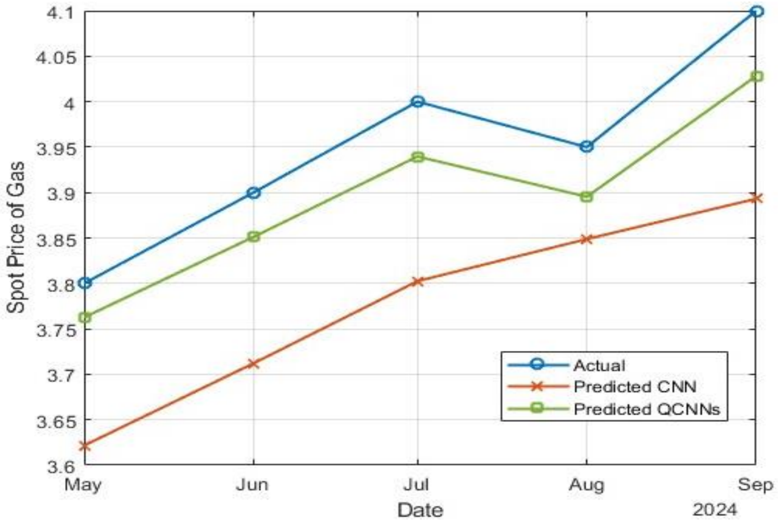

Figure 7: Actual and Predicted or Spot Price Gas with NNs and QCNNs without optimization algorithm

Table 3: Actual and Predicted Spot price of gas by NNs and QCNNs by Shark Smell Optimization

windows size	Actual	NNs	QCNNs	RMSE		Accuracy	
				NNs	QCNNs	NNs	QCNNs
1	3.80	3.77	3.80	0.16	0.17	0.95	0.96
2	3.90	3.86	3.90	0.10	0.09	0.96	0.98
3	4.00	3.96	3.99	0.13	0.11	0.95	0.98
4	3.95	3.98	3.94	0.10	0.07	0.97	0.98
5	4.10	4.06	4.08	0.20	0.15	0.95	0.99

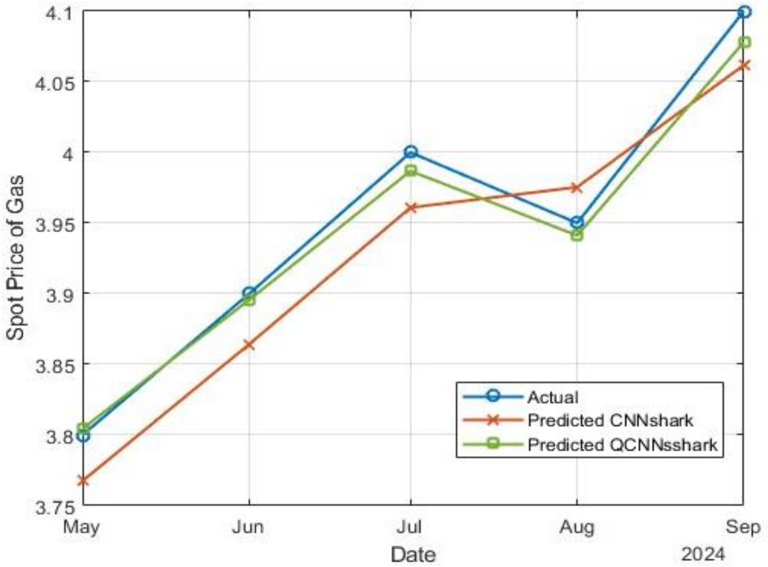

Figure 8: Actual and Predicted or Spot Price Gas with NNs and QCNNs by Shark Smell Optimization

9. Discussion of the results:

As for the performance analysis of NNs and QCNNs (the performance without an optimization algorithm shows that QCNNs outperformed NNs in most cases, as the accuracy of QCNNs ranged between 94% and 97%, while the accuracy of NNs ranged between 91% and 95%.. and the RMSE was lower in QCNNs, indicating expectations closer to the actual values. In the first window, the accuracy of QCNNs was higher (95%) compared to NNs (94%), with a lower RMSE () 0.13 versus 0.10. The performance of NNs was lower than QCNNs in all windows, especially in the fifth window, where the accuracy reached only 91%. As for the performance analysis after using the (Shark Smell Optimization) algorithm, it was noted that the overall performance of the algorithm improved, and an improvement was noted in the performance of both NNs and QCNNs: QCNNs accuracy became between 96% and 99%. And the RMSE decreased To 0.07 in some cases as in (Window 4). And NNs accuracy increased to 95%-97%. And RMSE decreased to 0.10-0.20. In the second window, RMSE was lower for QCNNs (0.09) compared to NNs (0.10), with higher accuracy (98%) versus (96%). This indicates that the performance improvement using SSO algorithm is attributed to improving the prediction process and reducing the gap between actual and predicted values. The graphs show the difference between actual and predicted values and the graph in Figure 7 shows the performance without optimization as the difference was clear between actual and predicted values for NNs. And the predictions using QCNNs were closer to the actual values. In Figure 8 with the use of the optimization algorithm, the gap between actual and predicted values decreased significantly, especially for QCNNs.

10. Conclusions

QCNNs outperform NNs in terms of error reduction and accuracy, both with and without optimization. Shark Smell Optimization algorithm significantly improved the performance of both types of networks. The results suggest that QCNNs with optimization algorithms could be a better choice for accurate predictions in similar applications.

References

[1] M. Wu, X. Liu, N. Gui, X. Yang, J. Tu, S. Jiang, and Q. Zhao, "Prediction of remaining time and time interval of pebbles in pebble bed HTGRs aided by CNN via DEM datasets," Nucl. Eng. Technol., 2022, https://doi.org/10.1016/j.net.2022.09.019.

[2] Dumakude, Aphelele ; E. Ezugwu , Absalom , (2023)," Automated COVID-19 detection with convolutional neural networks ", Scientifc Reports ,:10607 |https://doi.org/10.1038/s41598-023-37743-4

[3] Zhao, Xia ; Wang, Limin · Zhang, Yufei ; Han, Xuming ; Deveci , Muhammet , Parmar,Milan, (2024), " A review of convolutional neural networks in computer vision", Artifcial Intelligence Review (2024) 57:99 https://doi.org/10.1007/s10462-024-10721-6

[4] Fukushima, K.: Neocognitron, (1980). A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybern, 36, 193–202.

[5] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998). Gradientbased learning applied to document recognition. Proceedings of the IEEE, 86(11), pp.2278-2324.

[6] Yamashita , Rikiya; Nishio, Mizuho; Do , Richard Kinh Gian; Togashi, Kaori. (2018). Convolutional neural networks: an overview and application in radiology. Insights into Imaging (9:611–629 , pp:(1-19).

[7] Russakovsky O, Deng J, Su H et al (2015).Image Net Large Scale Visual Recognition Challenge. Int J Comput Vis , 115:211–252.

[8] Zhanga , Qingchen; T. Yang, Laurence ; Chenc , Zhiku; Lic , Peng . (2018). A survey on deep learning for big data . Information Fusion 42 , 146–157.

[9] Dhillon , Anamika ; K. Verma , Gyanendra. (2019). Convolutional neural network: a review of models, methodologies and applications to object detection . Springer, pp: 1-28

[10] Keiron O'Shea 1 and Ryan Nash, (2015). An Introduction to Convolutional Neural Networks . arXiv:1511.08458v2 [cs.NE].

[11] Baheti , Pragati , (2021) . A Comprehensive Guide to Convolutional Neural Networks . Automate repetitive tasks with V7's new Gen AI tool.

[12] TESSE, Riccardo; CHIABERGE, Marcello , (2022) . A Deep Learning approach to Instance Segmentation of indoor environment . POLITECNICO DI TORINO Master's Degree in MECHATRONIC ENGINEERING.

[13] Masood, Sarfaraz ; Rai , Abhinav ;Aggarwal , Aakash ; M.N. Doja; Ahmad, Musheer , (2020). Detecting distraction of drivers using Convolutional Neural Network. Pattern Recognition Letters

[14] Le Lu , Yefeng Zheng , Gustavo Carneiro, Lin Yang Editors, (2017). Deep Learning and Convolutional Neural Networks for Medical Image Computing :Precision Medicine, High Performance and Large-Scale Datasets . Springer, Library of Congress Control Number: 2017936345. 13

[15] ALBAWI , Saad; MOHAMMED, Tareq Abed; AL-ZAWI , Saad . (2017). Understanding of a Convolutional Neural Network .ICET2017, Antalya, Turkey. pp:1-7.

[16] Wang, Yingying ; Li, Yibin ; Song, Yong, ; Rong, Xuewen.(2020). The Influence of the Activation Function in a Convolution Neural Network Model of Facial Expression Recognition. Appl. Sci., 10, 1897; doi:10.3390/app10051897 www.mdpi.com/journal/applsci

[17] George , Gilbert ; Adeshina,Steve; Boukar , Moussa Mahamat. (2020). The Influence of Activation Functions in Deep Learning Models Using Transfer Learning for Facial Age Prediction . International Journal of Intelligent system and application in engineering,:2147-67992147-6799www.ijisae.org.

[18] Zafar , Afia ; Aamir, Muhammad ; Nawi , Nazri Mohd ; Arshad, Riaz , Ali, Saman; Alruban , Abdulrahman ; Dutta ; Kumar, Ashit and Almotairi ,Sultan, '(2022). A Comparison of PoolingMethods for Convolutional Neural Networks. Appl. Sci. 2022, 12, 8643. https:// doi.org/10.3390/app12178643

[19] Bhatt, D.; Patel, C.; Talsania, H.; Patel, J.; Vaghela, R.; Pandya, S.; Modi, K.; Ghayvat, H. CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope. Electronics 2021, 10, 2470. https://doi.org/10.3390/ electronics10202470

[20] Bao,Chenglong ; Shen, Qianxiao Li, Zuowei ; Wu, Cheng Tai, Lei ; Xiang, Xueshuang .(2014). Approximation Analysis of Convolutional Neural Networks. y Singapore MOE Research Grant MOE 2014-T2-1-065 and Tan Chin Tuan Centennial Professorship.

[21] Abedinia , Oveis ; Amjady, Nima , Ali Ghasemi, (2014). A new metaheuristic algorithm based on shark smell optimization . Weily , https://doi.org/10.1002/cplx.21634 15

[22] Magnuson, J.J. 4 Locomotion by scombrid fishes: Hydromechanics, morphology and behavior.(1979),. Fish Physiol, 7, 239– 313.

[23] Sfakiotakis, M.; Lane, D.M.; Davies, J.B.C.(1999). Review of fish swimming modes for aquatic locomotion. IEEE J Oceanic Eng. 1999, 24, 237–252

[24] Mohammad-Azari, S., Bozorg-Haddad, O., & Chu, X. (2017). Shark Smell Optimization (SSO) Algorithm. Studies in Computational Intelligence, 93–103. doi:10.1007/978-981-10-5221-7_10 14

[25] Karim , Pshtiwan J. ; Abdulhammed, Omar Y. ; Arif , Dashne R. ; Ali, Twana S. ; Abdalrahman, Avin O. ; Saffer, Arkan A. (2022). A secure image steganography based on modified matrix encoding using the adaptive region selection technique. Kurdistan Journal of Applied Research (KJAR) Print-ISSN: 2411-7684 | Electronic-ISSN: 2411-7706 Website: Kjar.spu.edu.iq | Email: kjar@spu.edu.iq

[26] Fawaz, Hassan Ismail; Germain Forestier ; Weber, Jonathan ; Idoumghar, Lhassane; Muller, Pierre-Alain, (2019). Deep learning for time series classification: a review. arXiv:1809.04356v4 [cs.LG], 1-44.

[27] https://tradingeconomics.com/commodity/natural-gas