AL- Rafidain University College

PISSN: (1681-6870); EISSN: (2790-2293)

Journal of AL-Rafidain University College for Sciences

Available online at: https://www.jrucs.iq

JRUCS

Journal of AL-Rafidain University College for Sciences

Comparison of CNN and SVM Approaches for Classifying Time Series of Caenorhabditis Elegans Motion

Omar A. Malaa	Osamah B. Shukur
omaromraan85@gmail.com	drosamahannon@uomosul.edu.iq
Department of Statistics and Informatics, College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq	

Article Information

Article History:
Received: February, 20, 2024
Accepted: April, 12, 2024
Available Online: 31, December, 2024

Keywords:
Caenorhabditis elegans (CE), Time series, Classification, Autoregressive (AR), Convolutional Neural Network (CNN), Support Vector Machine (SVM).

Correspondence:
Osamah B. Shukur
drosamahannon@uomosul.edu.iq

Abstract

Studying the motion of some roundworms types such as Caenorhabditis elegans (CE) is important to identify the actions and reactions and their effects of worm's life. In this study, the time series of CE motion represented by the angles of wave-motion between 1 to 177 degrees will be the case study. Each observation of this time series is a recorded frame (0.5 second) of 2.5 hours video of CE motion. A convolutional neural network (CNN) as one of deep learning techniques will be used to classify CE motion as dependent variable in binary cases based on the images of the angles of wave-motion as explanatory variable. The images of motion angles are imagined and designed by two dimensions image corresponding to every observation. These images combined into 4-d image (four dimensions matrix) to represent univariate explanatory variable. Support vector machine (SVM) will be also used to classify the angles of CE. In these types of data, the nonlinearity and uncertainty will be the most probably problems as reasons for in accurate classifications. CNN and SVM used with this type of dataset to improve the classification results. The results of comparisons explain that CNN approach outperforms SVM absolutely. In conclusion, CNN approach can be used to classify this type of time series with accurate results.

DOI: https://doi.org/10.55562/jrucs.v56i1.36

Introduction

The study of transparent nematodes in general is considered one of the important studies in microbiology, because their cells resemble human cells, as well as because of the rapid stages of their growth. There are many previous studies concerned with the movement of the worm, its speed, and the distances it travels during a specific period of time. The movement of the worm is consecutive during a specific time, in the form of a time series. As each movement that the worm makes or reaches is related to the movement before it to represent the study case for a single time series variable for the movement of the worm and it is within a period of time for fractions of a second for each movement made by the worm. In this study, data were obtained on Caenorhabditis elegans (CE) as an important typical organism in the study of genetics for a better understanding of behavioral genetics. The data includes observations, each one of them represents a specific angle of

Caenorhabditis elegans motion (CEM)[1]. Since CEM cannot be pinpointed, but it can be confined to periods, as it becomes necessary to classify it within categories, and then the attention is focused on classifying new observations through a classification model that is built through the behavior of the time series during the training period. The importance of classification lies in making predictions for data not seen in the training process and identifying the categories to which the new sample belongs. In this study, the time series classification study was addressed using supervised learning algorithms, which require the availability of target variable data to obtain learning errors.

CEM data is a long time series with a very large number of observations and a time proof that may be in seconds or fractions of seconds, which may give the character of non-linearity, which may make it difficult to deal with such data. Also, the very large length of the time series may be a major cause of heterogeneity, which results from the multiplicity of characteristics, qualities, and combinations that the data passes through from the beginning of the series to its end, which may make the results of point forecast inaccurate.

In order to reduce the problem of non-linearity and heterogeneity in the data and improve the prediction results, the data can be represented by images and binary classification which can be used as an alternative to the point forecast to improve the accuracy of the results compared to the prediction results with the presence of heterogeneity and non-linearity of the data.

There are previous studies dealing with the use of Convolutional Neural Network (CNN) and Support Vector Machine (SVM) in the field of microbiology regarding the behavior of nematodes. The researcher [1] used the CNN method to classify microscopic images to identify a specific type of nematode. [2] also used the CNN method to distinguish and classify genetically diverse CE strains by training the model on time series data of worm positions using samples of worm movement images as an input variable. [3] also used deep learning through CNN and Recurrent neural network (RNN) methods to calculate the life expectancy of CEM data by classifying them as alive or dead by observing images of the worm's movement.

[4] used the SVM method and two other methods for binary classification by testing a data set to identify a skin disease caused by a fungal infection of segmented worms, so the classification of the disease is infected or healthy for each case. Also [5] chose to use SVM for binary classification because of the power of this classifier to identify one of the important parts of the CE structure, which is the head of the worm.

Materials and Methods

framework of study

The framework of this study includes the following:

a. Determining the suitable AR model.
b. Constructing the appropriate CNN.
c. Constructing the appropriate SVM based on AR models.
d. Calculating the accuracy measurements for CNN and SVM classifications.
e. Comparing the classification results to determine which model would be provided better classification accuracy.

Auto-Regressive Model.

Time series is a set of observations generated at consecutive periods of time distinguished by lack of independence. As the observations in it are related to its predecessor chronologically, through which it is possible to predict future time series based on observations of a time series that occurred in the past [6]. The current time series can be expressed using the autoregressive function for the values of the previous time series, and the autoregressive function rank p can be written as in the equation below.

[1] Archive of the UEA&UCR time series classification available to the public of researchers: http://www.timeseriesclassification.com/description.php?Dataset=EigenWorms data downloaded at 12-09-2022

$$\begin{aligned}&\phi(B)x_t = e_t\\&\rightarrow (1-\phi_1 B-\phi_2 B^2-\cdots-\phi_p B^p)x_t = e_t\\&\rightarrow x_t = \phi_1 x_{t-1}+\phi_2 x_{t-2}+\cdots+\phi_p x_{t-p}+e_t\end{aligned} \quad (1)$$

where ϕ_r is the r^{th} auto-regressive coefficient, x_{t-r} on x_t in the auto-regressive model, $r = 1,2,3,\ldots,p$, e_t is the random error or the white noise with mean zero and constant variable σ_e^2 where $e_t \sim i.i.d.N(0,\sigma_e^2)$, and $\phi(B) = (1-\phi_1 B-\phi_2 B^2-\cdots-\phi_p B^p)$.

$W_t = (1-B)^d x_t$; W_t is after differencing to satisfy the mean stationary and will be written instead of x_t in the equation **Error! Reference source not found.**. With autoregressive models, the Box-Jenkins methodology is used with its four steps: identification, parameter estimation, diagnostic checking, and forecasting.

CNN

It is one of the basic tools for deep learning, which falls under the umbrella of deep neural networks (DNN), which includes another type of DNN, which is RNN. The CNN method is sometimes called the Deep-CNN, when it is a multi-layered network that contains more than two layers, and since the CNN method currently has a structure of two layers or more, the terms CNN and Deep-CNN have the same scientific concept [7]. The CNN structure consists of two main parts, the feature recognition layer in which the convolution and pooling operations are performed to recognize image properties such as edges and gradation, and the fully connected layer that receives the output of the feature recognition layer as input to be classified as shown in

Figure 1 below.

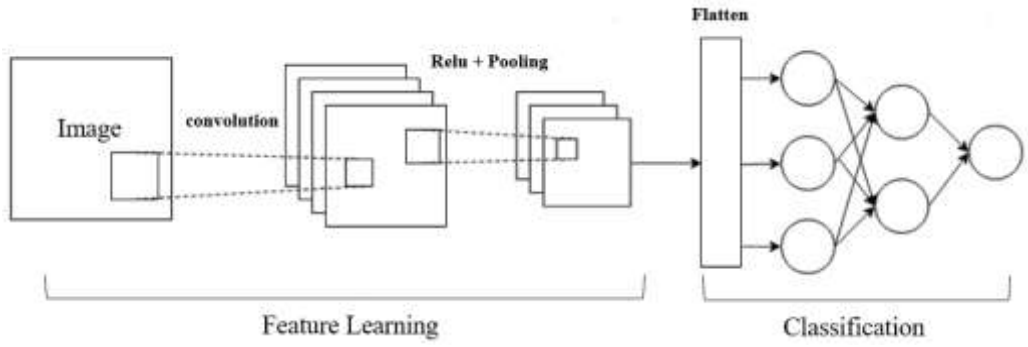


Figure 1: Structure of CNN

First, the appropriate CNN structure is chosen by specifying the number of layers. Layers consist of an input layer, hidden layers, and an output layer. As for the input layer, it is determined by the number of inputs, which correspond to the concept of explanatory variables in the multiple linear regression model, or they are called autoregressive variables for the time series. The size of the hidden layer is determined by the number of neurons it contains, which is often more than the number of inputs. As for the output layer, the number of outputs matches the number of

classification options required, since in the binary classification one output is sufficient, whose value will be equal to (1 or -1) according to the classification of the image that we have. It can be noted that the higher the number of hidden layers, the higher the quality and efficiency of the network, and thus the time allotted to it.

The sub-areas of the input images are connected to the first hidden layer, as in this layer the filter size is specified (f) and the number of Filters (m) that wraps around all points of the image to identify edges, vertical or horizontal lines, or other angles of the image and the details that form the picture. After that, the initial values of the weights are randomly determined, symbolized by (W) in the filters, which are small random numbers, as the small numbers speed up the calculation process to reach the appropriate weights. Filter weights help to determine the importance of any variable as each input image is multiplied by a matrix of filter weights.

When wrapping the filter around the image, the size of the image will shrink and many data and image properties will be lost. To solve this problem, Padding is added with rows and columns along the width and the length of the image and its perimeter.

As for the stride in which the filter moves (meaning the number of pixels to be skipped), if it is not specified, it will be default (1) that is, one step for the filter towards the left of the image.

The size of the output resulting from the process of wrapping around the image is obtained depending on the size of the input, the size of the filter, the padding and the number of strides by a simplified mathematical process as shown in equation **Error! Reference source not found.** below.

$$I \times J = \left[\left(\frac{H+2p-f}{s}\right)+1\right] \times \left[\left(\frac{W+2p-f}{s}\right)+1\right] \quad (2)$$

Since H : the length of the image, W (width of the image, p : padding, f : one of the dimensions of the filter, s : stride. Then each bias value is summed with each element of the matrix that corresponds to it to get the output of the convolution process as in the equation below whose output can be called the output of the additive function which represents the first stage of the hidden layer.

$$SUM_k = \sum_{i=1}^{f_2} \sum_{j=1}^{f_1} w_{f_1 f_2} x_{HW} + b_k \quad (3)$$

When i, j represent dimensions or number of rows and the number of columns in each image respectively. And that f_1, f_2 are the number of rows and columns in each filter respectively. And that $\mathrm{k} = \mathbf{1}, \mathbf{2}, \ldots, \mathrm{m}$ which represents the series result of each filter.

One of transfer functions within the hidden layer can be applied on the outputs of convolution process, one of the most using transfer functions in neural network is as following.

1. Log-Sigmoid function:

$$f\left(\mathrm{SUM}\right) = \frac{\mathbf{1}}{\mathbf{1}+e^{-(\mathrm{SUM})}} \quad (4)$$

As SUM represents the input of the transfer function which represents the output of the additive function and generates output within the limits (0,1) and as shown in **Error! Reference source not found.**.

2. Tan sigmoid function:

$$f(\mathrm{SUM}) = \frac{\mathbf{2}}{\mathbf{1}+e^{-2(\mathrm{SUM})}} - \mathbf{1} \quad (5)$$

As SUM represents the input of the transfer function which represents the output of the additive function and generates output within the limits (-1,1) and as shown in **Error! Reference source not found.** below.

3. Pure Line function:

$$f(\mathrm{SUM}) = \mathrm{SUM} \quad (6)$$

As SUM represents the inputs of the transfer function which represents the outputs of the

additive function and generates outputs within the limits (-1,1) and as shown in **Error! Reference source not found.** below.

4. Rectified Linear Unit function: It can be defined as follows.

$$f(\text{SUM}) = \begin{cases} \mathbf{0} & \text{SUM} \le \mathbf{0} \\ \boldsymbol{x} & \text{SUM} > \mathbf{0} \end{cases} \quad (7)$$

Since x represents the inputs of the transfer function which represents the outputs of the additive function and generates outputs greater than or equal to (0) and as shown in **Error! Reference source not found.** below.

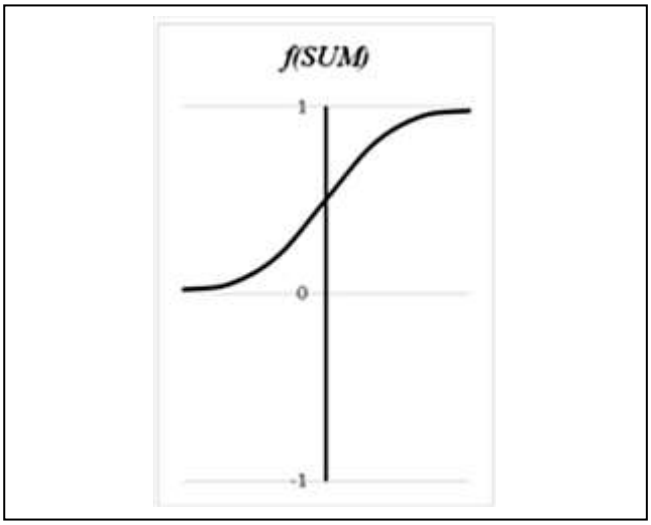


Figure 2: logistic sigmoid function

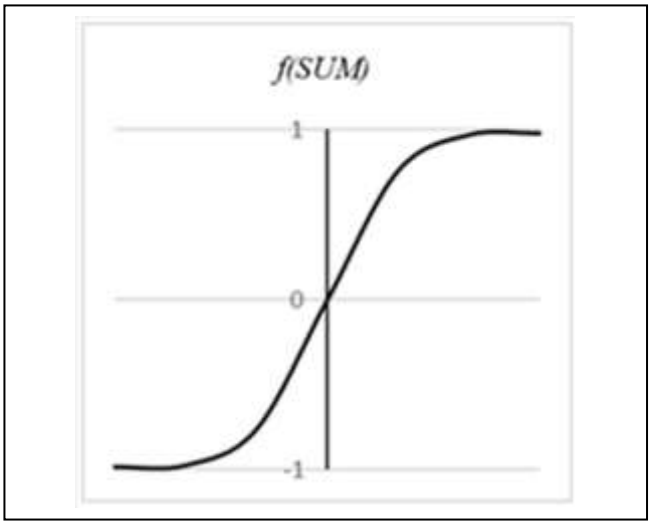


Figure 3: tan sigmoid function

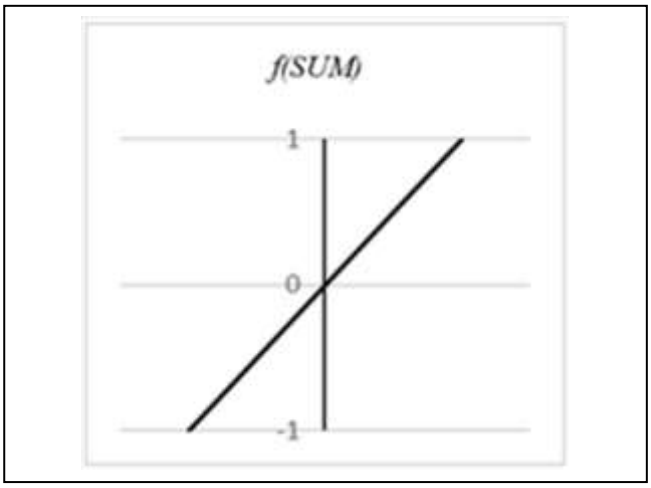


Figure 4 : Pure Line function

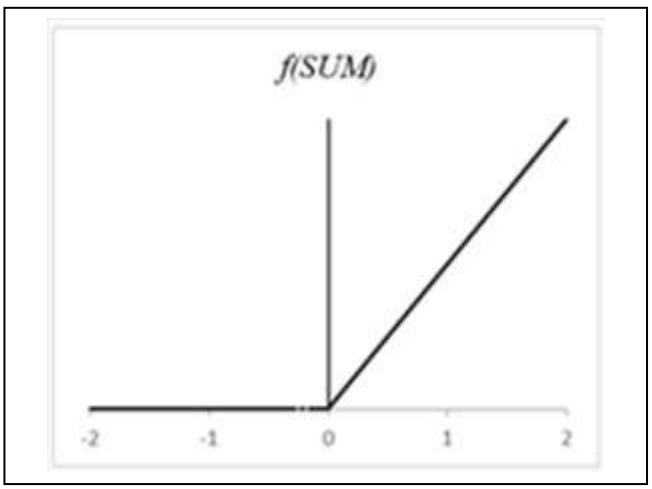


Figure 5: ReLU function

After the process of introducing the transfer function to the outputs of the additive function, the pooling process is applied by dividing the matrix into square or rectangular compilation areas, as the matrix is divided into a number of small matrices, and this is done by determining its size and number of steps. Then the highest value is taken for each area in case of applying max pooling, or

the average of the values in case of average pooling. Thus, each small matrix is reduced to only one value, which results in reducing the size of the matrix so that the matrix becomes at the end of the first hidden layer with fewer dimensions, then it is converted into a vector, which was obtained through a process called flatten. The vector represents the inputs of the Fully Connected Layer, and this layer combines all the features learned by the previous layers through images [8]. In the regression layer, the error rate is calculated using the MSE. The purpose of including the regression layer is to reduce the error by updating the weight value and biased value, which is an iterative process until the optimal values are reached, as each iteration process requires obtaining certain values of weights and bias and testing the error rate, this process is called stepwise regression, this procedure is called Gradient Descent (GD), which allows determining the direction towards reducing errors. The GD equation is as follows.

$$p_{i+1} = p_i - \alpha \sum_{j=1}^{n} (y_{ij} - \hat{y}_{ij}).x_{ij} \quad (8)$$

As (α) represents the Learn Rate, which is a very small value confined between (0 and 1), and p_i represents w_i or b_i in the current iteration, and p_{i+1} represents w_{i+1} or b_{i+1} in the new subsequent iteration, y_{ij} represents the value of observation j of the target variable in the current iteration i, $\hat{y}_{ij}$ represents the value of observation j of the output variable in the current iteration i of the regression layer, x_{ij} represents the observation j of the input variable in the current iteration i [9]. After obtaining the optimal values for the weights and biases through the regression layer, the final procedure that CNN takes is to classify the time series by entering a function on it, this function performs a process called (threshold) as shown in Equation **Error! Reference source not found.** below whereby any observation is converted from the output variable of the regression layer and the final output variable is created.

$$\hat{y}_{\text{final}} = \begin{cases} -1 & \hat{y} < 0 \\ +1 & \hat{y} > 0 \end{cases} \quad (9)$$

By comparing the variable $\hat{y}_{\text{final}}$ with the original values variable, which is the target variable, the accuracy of the classification model will be calculated for the real values of the time series using the evaluation scale of classification accuracy. The

Figure 6 shows the general framework of the CNN algorithm.

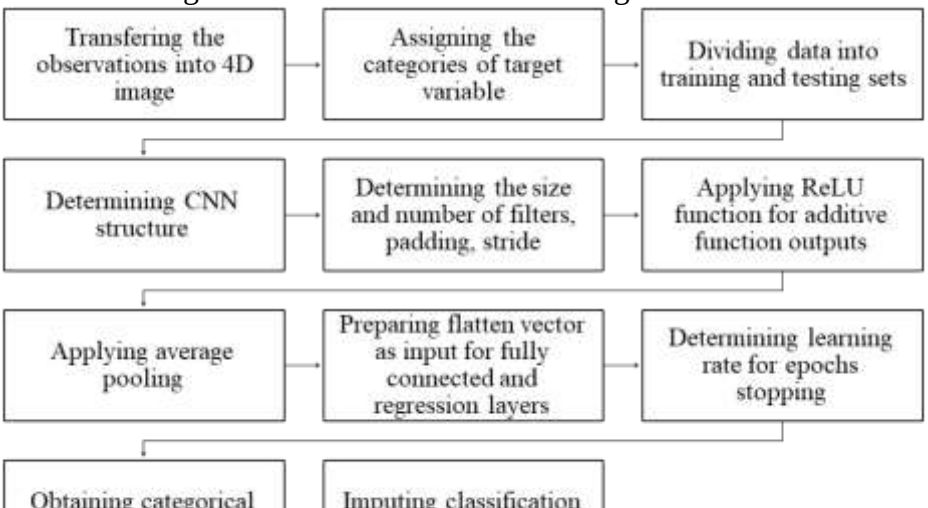


Figure 6: general structure for CNN algorithm

Support vector machine (SVM)

The idea of the support vector machine (SVM) searches for the optimal level that separates the data into two categories, meaning that the observation points on both sides of the line represent two different categories separated by a dividing line called the hyperplane. These two categories are often positive and the other negative, as the hyperplane has two parallel margins at the closest points which are called support vectors, as shown in . SVM is a supervised machine learning method which has various applications include face recognition, image classification, handwriting recognition, and many others. The hyperplane can be represented as follows:

$$w.x+b=0 \tag{10}$$

The supporting vector of the positive and negative category is represented as in the two equations, respectively:

$$w.x_2+b=1 \tag{11}$$

$$w.x_1+b=\text{-}1 \tag{12}$$

As $w=\{w_1,w_2,\ldots,w_d\}$ a vector controls the direction of the hyperplane and is orthogonal to it and $d=\{(x_1,y_1),(x_2,y_2),\ldots,(x_p,y_p)\}$, d represents the number of lags of the time series corresponding to the Auto Regression variables, b is the bias that controls the distance between the hyperplane and the point of origin, that is, it represents the intercept constant with the axis y, x which are the time lags, that are, the influential autoregressive variables.

The sum of the two vertical distances of two support vectors on both sides to the hyperplane is called margin, and the purpose of this algorithm is to obtain the largest margin to achieve the optimum level with the least misclassifications, according to the following equation.

$$\max_{w,b}=\frac{w}{\|w\|}(x_2-x_1)=\frac{2}{\|w\|} \tag{13}$$

Since $\|w\|$ represents the size of the vector *w* and it is also called (Norm) and equation **Error! Reference source not found.** is to obtain the largest margin and find the factors *w* and b , the margin can be improved by rewriting the equation in the following form.

$$\min_{w,b}\frac{1}{2}\|w\|^2 \tag{14}$$

To solve this problem, it is necessary to enter Lagrange's Multipliers λ_i as follows.

$$L(w,b,\lambda)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{L}\lambda_i y_i(x_i.w+b)+\sum_{i=1}^{L}\lambda_i \tag{15}$$

If the data is not linearly separable, then the division process takes place by adding an additional dimension to the data so that the hyperplane can separate the categories, this method is called Kernel, the task of this function is data segregation. There are several types of kernel functions, and we mention one of them [10].

Gaussian Kernel function (GKF):

$$K(x_i,x_j)=\exp(-\frac{\|x_i-x_j\|^2}{2\sigma^2})=\exp(-\sum_{j=1}^{n}\frac{(x_i-x_j)^2}{2\sigma^2}) \tag{16}$$

The GKF function determines the amount of similarity and difference between the inputs, which means that it divides the data into two parts for the purpose of classifying it, as (x_i) is the value of the point that is tested or compared with the value of the point (x_j), that is, when the values of (x_i, x_j) are close to each other, the difference is small and close to zero, and thus everything inside the exponential function is close to zero, and the result becomes 1. When these values are far from each other, the difference is large, and therefore there is a negative large number inside the exponential function, and the result is zero or close to zero, since that (x_i) and (x_j) are two vectors for the dimension p in the matrix k for the observations i and j in the variable x [11]. The Figure 7 shows the general framework of the SVM algorithm.

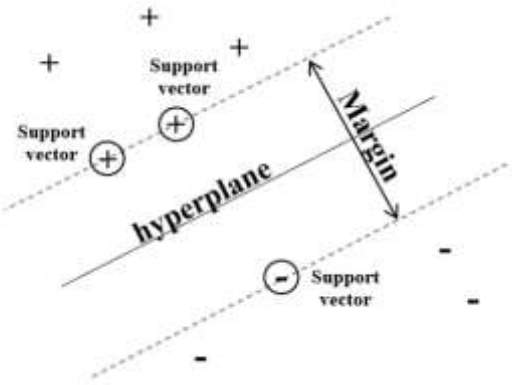


Figure 6 : Separating the space linearly and showing the margin

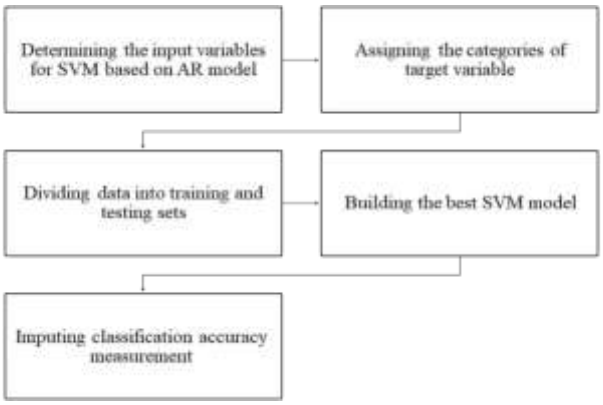


Figure 7 : general structure for SVM algorithm

Accuracy Measurement

These scales are used to measure the accuracy of the model's performance in the classification. In order to identify these measures, one must know the confusion matrix [12], as shown in Table 1.

Table 1 : Confusion Matrix

		Actual	
		Yes	**No**
Predicted	**Yes**	True positive (TP)	False positive (FP)
	No	False negative (FN)	True negative (TN)

One of the simplest measures used in classification is the Accuracy scale, in which the ratio of expected cases matching actual cases to the total number of all expected and actual matching and non-conforming cases is calculated as follows.

$$\text{Accuracy} = \frac{\text{Number of correctly classified}}{\text{Total Number}} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \times 100 \quad (17)$$

To calculate the inaccuracy of the Accuracy scale, it is done by subtracting the result of equation **Error! Reference source not found.** from the number 1, as follows.

$$\text{Inaccuracy} = 1\text{-Accuracy} = \frac{\text{FP+FN}}{\text{TP+TN+FP+FN}} \times 100 \quad (18)$$

Results and Discussion

Data used in the study

data is the shadow of CEM corners in a bacterial food garden on an agar plate as long time series with too many observations. CEM was determined by tracking using videos split as frames at a rate of at least 2 frames per second. The time series is a video of the worm's movement over

approximately 2.5 hours. As the angle of movement determines the speed of movement, so the angle of curvature of the worm's body is used to indicate its speed, the more the angle is sharp, the speed increases, so the worm travels a greater distance in less time. The degree of angles of the CE body ranges from 1◦ when it bends almost on itself to an angle of less than 180◦ as the largest possible angle when it reaches almost straightness.

Speed is defined as the distance between the midpoints of the start frame and the end frame divided by the time between both frames. The average speed of CEM is estimated at least 5% of its length per tire, as the worm must maintain this speed continuously, compensating for the delay caused by stopping for some reason by adjusting the speed by increasing its speed by increasing the intensity of the angles of movement until the required speed rate is reached [13].

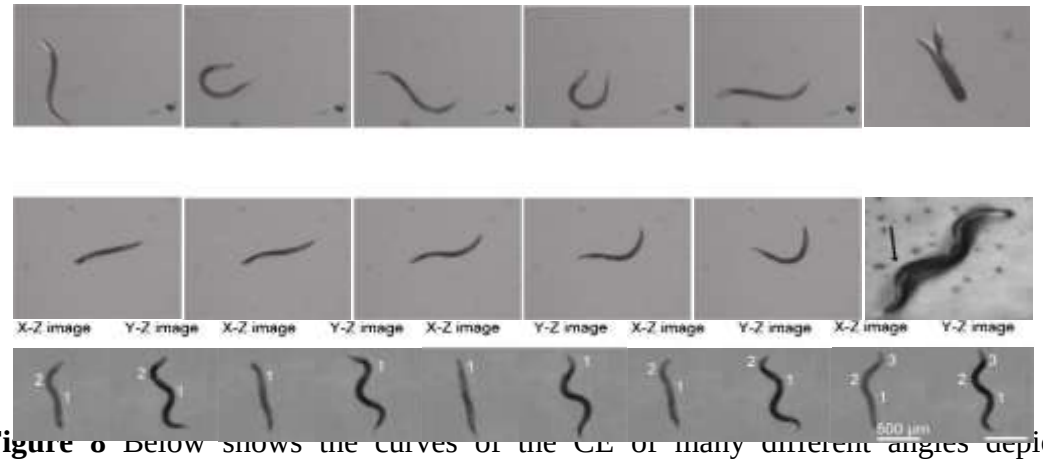


Figure 8 Below shows the curves of the CE of many different angles depicted from different dimensions.

Figure 8 : CE swimming and motion in different angles and dimensions [14].

In certain studies, the interest is in studying when the worm travels greater distances in less time, when it moves quickly, that is, when it moves in sharp angles. There are other studies in which the concern is about the stops of the worm due to a problem or a slowness of its movement, that is, when its movement is at relatively obtuse angles. Therefore, the positive quality and the negative case are determined according to the nature of the study. And because it is difficult to predict this number of numerical values for the CEM angles because the values are many and close to each other, but these angles can be classified categorically according to the speed of movement into fast movement with sharp angles that represent the positive quality (+1) and slow movement with obtuse angles that represent the negative quality (-1), binary classification. In this study, it has been relied on transferring the degrees of CEM angles into graphic forms through two-dimensional grayscale images, as shown in

Figure 9 below.

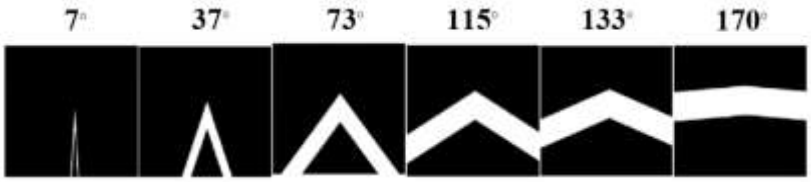


Figure 9 : samples of transferring degrees of CEM angles into graphic forms

The number of CEM time series observations was (17984) for five strains (the N2 reference strain, goa-1, unc-1, unc-38 and unc-63). Two time series were randomly selected from each strain and each time series represented the CEM of a single CE worm. The two-dimensional images represent the independent variable (*x*). These images were classified into two categories (1 and -1), which are acute and obtuse, for the purpose of representing the dependent variable (*y*).

The CNN method is particularly suitable for analyzing image data of the angle shapes formed by the CE when it moves in the form of observation per unit time within the range approximately (1˚) to approximately (177˚). The threshold limit between acute and obtuse angles is angle (90) for binary classification.

To create the input variable for the CNN method, each time series was converted from its numerical format and formed into two-dimensional images using the MATLAB program, by drawing the angle that takes the range (1-177), and then saving each time series variable as a four-dimensional matrix. The first and second dimensions represent the dimensions of each image for the angle corresponding to each view. As for the fourth dimension of the image, it represents the sequence of viewing that was expressed as an image. In this study, CEM images appeared automatically with a size of (246×251) pixels, with a number of 14400 views, equivalent to approximately 80% of the total views, and their numbers 17984 views corresponding to the training period, and 3584 views, approximately 20% of the total views for a period, therefore, the final size of the input variable image is (246×251×1×17984) pixels.

CNN

The general framework of the CNN implementation algorithm includes the implementation of several sequential steps as follows.

1. Converting the observations from their digital state into a single four-dimensional matrix that collects the images together.
2. Determining the positive and negative categories of the target variable in two qualities, acute and obtuse angles.
3. Dividing the time series observations into two groups for training and testing.
4. Determining the structure of the convolutional neural network, the input layer, the hidden layer (2), and the output layer, meaning that the number of layers in general is (1-2-1).
5. Determining the size of the filter (3×3), the number of filters (8), the padding (3×3), and one stride in the first hidden layer, as is common in previous works.
6. Figure **10** shows the size of the image, the size of the filter, and the process of wrapping the filter on a part of the image.
7. Add the bias value with each element of the matrix (250×255×8) resulting from step 5 and equation **Error! Reference source not found.** and apply the ReLU function to it.
8. Average Pooling of size (2×2) and step (2) were applied to the result of the ReLU function, which results in the size of the output variable with dimensions (125×127×8).
9. Combining the outputs of average pooling in one column by a process called Flatten. A vector of size (127000×1) is obtained representing the Fully Connected Layer to which the

Regression Layer is connected.

10. After determining the learning rate (0.01), the process was stopped at the eighth iteration after obtaining the required learning rate.

11. Converting all-time series values of the regression layer output variable according to equation **Error! Reference source not found.**.

12. Measuring the accuracy of the classification model by applying equation **Error! Reference source not found.**.

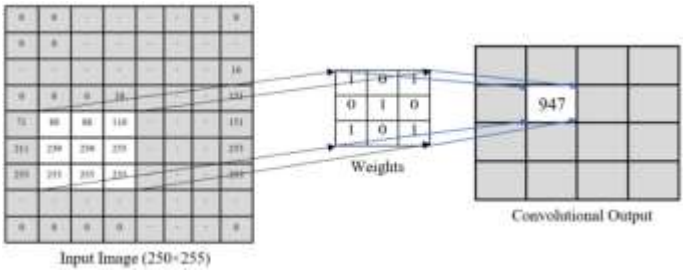


Figure 10: the size of filter with convolution for part of image.

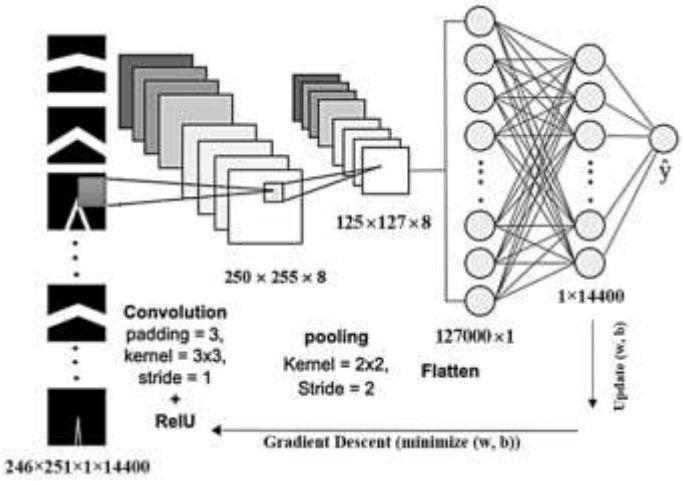


Figure 11 : CNN Algorithm

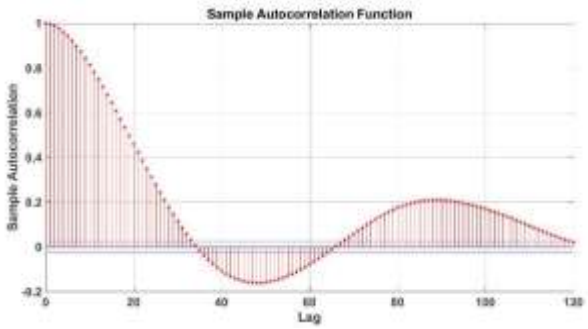
Figure **11** above shows the convolutional neural network algorithm that was applied to the motion angle images of the worm. The results of measuring the accuracy of the classification model for the training and testing data are as in Table 2 below.

Table 2 : The results of classification model accuracy

	First strain	Second strain	Third strain	Forth strain	Fifth strain
Training data	100%	100%	100%	100%	100%
Testing data	100%	100%	100%	100%	100%
Training data	100%	100%	100%	100%	100%
Testing data	100%	100%	100%	100%	100%

AR

Depending on the principle of autoregressive and autocorrelation, the rank of the Auto Regressive (AR) model was determined through the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF), regardless of the stability of the data, it is possible to deduce that the best autoregressive model for the first strain sample, as in Figure 12 below is AR(5), because ACF gives a gradually decaying pattern with slow decay, which indicates instability. While PACF gives a pattern of sudden interruption after (5) time lags as in Figure 13.



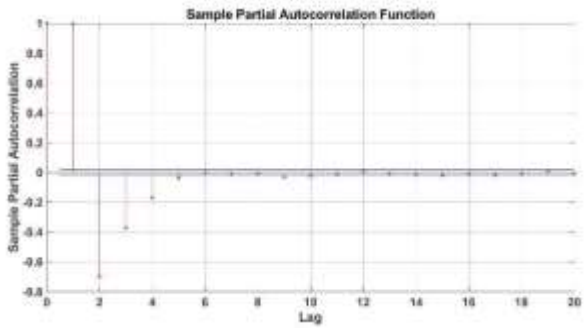


Figure 12: (ACF) Autocorrelation of the first strain sample

Figure 13 : (PACF) Partial Autocorrelation of the first strain sample

Accordingly, after applying the ACF autocorrelation function on the training data, the time lags referred to in Table 3 below were used.

Table 3 : The most appropriate autocorrelation ranks

	First strain	Second strain	Third strain	Forth strain	Fifth strain
First sample	5	5	7	8	8
Second sample	5	5	6	7	2

SVM

The general framework of the SVM implementation algorithm includes the implementation of several sequential steps as follows.

1. Using the optimal autoregressive variables based on Table 3 to determine the input variables for the SVM method.
2. Determining the positive and negative categories of the target variable with two qualities, acute and obtuse angles.
3. Dividing the time series observations into two groups, training and testing.
4. Building the best SVM model using the training data for the purpose of binary classification and the kernel function used is as in equation **Error! Reference source not found.** using the directive (fitcsvm(XTrain,YTrain)) in MATLAB program.
5. Using the model in the previous step to classify the data in the test period using the directive (predict(SVMModel2,XTest)) in the MATLAB program..
6. Measuring the accuracy of the classification model by applying equation **Error! Reference source not found.**.

The results of measuring the accuracy of the classification model for the training and testing data are as in Table 4 below.

Table 4 :Classification accuracy of the five strains of training and test data using SVM.

	First strain	Second strain	Third strain	Forth strain	Fifth strain
Training data	99%	97%	98%	98%	99%
Testing data	97%	97%	99%	98%	99%
Training data	%99	97%	%99	99%	%99
Testing data	98%	98%	98%	99%	99%

Conclusions

In this study CNN method is used which concerns about transferring digital observations to images as suggested method to improve the results of classification accuracy of time series data for CE worm. Two samples of data are used each one of them contains five strains and the results show overpassing the suggested method CNN on SVM as an alternative classification method when using AR as a main source to determine the number of input variables to SVM. Scale classification accuracy is used to show the quality of classification. Using CNN method as optimum way can be concluded with time series data after transferring its observations to images for one of the roundworms types which have too many time series observations.

References

[1] J. Uhlemann, O. Cawley, and T. Kakouli-Duarte, "Nematode Identification using Artificial Neural Networks", in DeLTA, 2020.

[2] A. Javer, A.E. Brown, I. Kokkinos, and J. Rittscher, "Identification of C. elegans strains using a fully convolutional neural network on behavioural dynamics", in Proceedings of the European Conference on Computer Vision (ECCV) Workshops, 2018.

[3] A. García Garví, J.C. Puchalt, P.E. Layana Castro, F. Navarro Moya, and A.-J. Sánchez-Salmerón, "Towards lifespan automation for Caenorhabditis elegans based on deep learning:

analysing convolutional and recurrent neural networks for dead or live classification", Sensors, **21**(14): p. 4943, 2021.

[4] S. Kundu, N. Das, and M. Nasipuri, "Automatic detection of ringworm using local binary pattern (LBP)", arXiv preprint arXiv:1103.0120, 2011.

[5] M. Zhan, M.M. Crane, E.V. Entchev, A. Caballero, D.A. Fernandes de Abreu, Q. Ch'ng, and H. Lu, "Automated processing of imaging data through multi-tiered classification of biological structures illustrated using Caenorhabditis elegans", PLoS computational biology, **11**(4): p. e1004194, 2015.

[6] P.J. Brockwell and R.A. Davis, "Time series: theory and methods", Springer science & business media, 2009.

[7] O. Theobald, "Machine learning for absolute beginners: a plain English introduction"; Vol. 157. Scatterplot press, 2017.

[8] R.E. Neapolitan and X. Jiang, "Artificial intelligence: With an introduction to machine learning", CRC Press, 2018.

[9] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional neural networks for time series classification", Journal of Systems Engineering and Electronics, **28**(1): p. 162-169, 2017.

[10] A. Ng, "CS229 Lecture notes", CS229 Lecture notes, **1**(1): p. 1-3, 2000.

[11] K.-L. Du and M.N. Swamy, "Neural networks and statistical learning", Springer Science & Business Media, 2013.

[12] A. Luque, A. Carrasco, A. Martín, and A. de Las Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix", Pattern Recognition, **91**: p. 216-231, 2019.

[13] E. Yemini, T. Jucikas, L.J. Grundy, A.E. Brown, and W.R. Schafer, "A database of c. elegans behavioral phenotypes", Nature Methods, **10**(9): p. 877–879, 2014.

[14] A. Bilbao, A.K. Patel, M. Rahman, S.A. Vanapalli, and J. Blawzdziewicz, "Roll maneuvers are essential for active reorientation of Caenorhabditis elegans in 3D media", Proceedings of the National Academy of Sciences, **115**(16): p. E3616-E3625, 2018.

مجلة كلية الرافدين الجامعة للعلوم (2024)؛ العدد 56؛ 398- 410

AL- Rafidain University College

PISSN: (1681-6870); EISSN: (2790-2293)

مجلة كلية الرافدين الجامعة للعلوم

Available online at: https://www.jrucs.iq

JRUCS

Journal of AL-Rafidain University College for Sciences

مقارنة بين اسلوبي CNN و SVM لتصنيف السلسلة الزمنية لحركة الربداء الرشيقة

عمر اكرم محمد سعيد	أ.م.د. أسامة بشير شكر
omaromraan85@gmail.com	drosamahannon@uomosul.edu.iq

قسم الاحصاء والمعلوماتية، كلية علوم الحاسوب والرياضيات، جامعة الموصل، الموصل، العراق

المستخلص

من المهم دراسة حركة بعض أنواع الديدان الأسطوانية مثل دودة الربداء الرشيقة Caenorhabditis elegan (CE) للتعرف على الأفعال وردود الفعل وتأثيراتها على حياة الدودة. في هذه الدراسة، ستكون السلسلة الزمنية لحركة CE الممثلة بزوايا الحركة الموجية بين 1 إلى 177 درجة هي دراسة الحالة. كل ملاحظة لهذه السلسلة الزمنية عبارة عن إطار مسجل (0.5 ثانية) لفيديو مدته 2.5 ساعة لحركة CE. سيتم استخدام الشبكة العصبية التلافيفية convolutional neural network (CNN) كأحد تقنيات التعلم العميق لتصنيف حركة CE كمتغير تابع في الحالات الثنائية بناءً على صور زوايا الحركة الموجية كمتغير توضيحي. يتم تصور وتصميم صور زوايا الحركة بواسطة صورة ذات بعدين تتوافق مع كل مشاهدة. تم دمج هذه الصور في صورة رباعية الأبعاد (مصفوفة ذات أربعة أبعاد) لتمثيل متغير توضيحي أحادي المتغير. سيتم أيضًا استخدام آلة المتجه الداعم support vector machine (SVM) لتصنيف زوايا CE. في هذه الأنواع من البيانات ستكون اللاخطية وعدم اليقين هي المشاكل الأكثر احتمالا كأسباب في التصنيفات الدقيقة. يتم استخدام CNN وSVM مع هذا النوع من مجموعات البيانات لتحسين نتائج التصنيف. توضح نتائج المقارنات أن اسلوب CNN يتفوق على SVM تمامًا. في الختام، نستنتج انه يمكن استخدام اسلوب CNN لتصنيف هذا النوع من السلاسل الزمنية بنتائج دقيقة.

معلومات البحث

تواريخ البحث:

تاريخ تقديم البحث:20/2/2024
تاريخ قبول البحث:12/4/2024
تاريخ رفع البحث على الموقع:
31/12/2024

الكلمات المفتاحية:

Caenorhabditis elegans (CE)، السلاسل الزمنية، التصنيف، الانحدار الذاتي (AR)، الشبكة العصبية التلافيفية (CNN)، آلة ناقل الدعم.(SVM)

للمراسلة:

أ.م.د.أسامة بشير شكر
drosamahannon@uomosul.edu.iq

DOI: https://doi.org/10.55562/jrucs.v56i1.36