Received: 11 December 2023 | Revised: 1 April 2024 | Accepted: 8 May 2024 | Published online: 15 May 2024

RESEARCH ARTICLE

Artificial Intelligence and Applications
2024, Vol. 2(4) 271–280
DOI: 10.47852/bonviewAIA42022277

BON VIEW PUBLISHING

# Soliton Solutions of Some Ocean Waves Supported by Physics Informed Neural Network Method

**Ismail Onder[1], Abdulkadir Sahiner[1], Aydin Secer[2] and Mustafa Bayram[2,*]**

[1]*Mathematical Engineering, Yildiz Technical University, Turkey*

[2]*Computer Engineering, Biruni University, Turkey*

**Abstract:** In this study, we aim to obtain numerical results of the modified Benjamin-Bona-Mahony equation, Ostrovsky-Benjamin-Bona-Mahony equation, and Mikhailov-Novikov-Wang equation via the physics-informed neural networks (PINN) method. The equations are modeled for shallow and long water waves, as well as fundamental and phenomenal models in ocean engineering. According to the implementation, we obtained the PINN solutions of kink, bright, multi-soliton (two-soliton), and mixed dark-bright soliton solutions. According to the inference from the obtained results, we achieved good results in some cases compared to other approximate solution methods in the literature. However, it was also observed that the best possible results could not be obtained in cases where the soliton type was intricate and layered. While the results were obtained, the number of hidden layers and the number of neural networks in the layers also varied. These results are shown in tables. Since it is known that the aforementioned models are not solved by the PINN method, we anticipate that the study will lead to other studies in the field of ocean engineering.

**Keywords:** soliton solutions, modified Benjamin-Bona-Mahony equation, Ostrovsky-Benjamin-Bona-Mahony equation, Mikhailov-Novikov-Wang equation, physics-informed neural networks

## 1. Introduction

Machine learning and artificial neural networks draw attention to the advantages they provide and the widening of their usage areas. Machine learning and artificial neural networks, which are extensively used in scientific studies, are helpful technologies for researchers to clarify time and focus. One of these fields is differential equations [1, 2].

Differential equations are used effectively in water wave models. This situation is becoming an area of interest for researchers in the integration of machine learning and artificial neural networks, which are today's technologies, into these areas. The physics-informed neural networks (PINN) model proposed by Raissi et al. [3] to solve nonlinear PDEs can also be evaluated in this context. The PINN model, which is one of the models with increasing potential in mathematical physics and engineering, was used in this study with the support of artificial neural networks in physically based solutions of ocean waves.

Some of the studies used by PINN to obtain numerical solutions are as follows: The PINN method was discussed by Fang et al. [4] within the scope of different soliton solutions of the nonlinear Schrodinger equation (NLSE). On the other hand, PINN method is used to examine the propagation of solitons in water depending on the KdV equation [5]. In a different study, Wu et al. [6] investigated the standard NLSE with the PINN method based on artificial neural networks they developed and conducted studies on estimating the coefficients. In the study by Peng et al. [7], investigations were made on a rigged periodic wave, breath wave, soliton wave, and periodic wave solutions using the Chen-Lee-Liu equation. In Lin and Chen [8], Sawada-Kotera equations were analyzed with the PINN method. Also, there are some different versions of the PINN method in the literature, such as multiple parallel subnets PINN [9], improved PINN [10], coupled automatic-numeric PINN (CAN-PINN) [11], and energy conservation deep learning method [12]. On the other hand, Esen et al. [13] and Onder et al. [14] also find some recent studies about obtaining soliton solutions to wave equations. The method is a machine learning technique, and since it is quite new, it also raises some questions. One of these is the stability and convergence analysis. Unfortunately, the subject has not been examined in this study, but readers can take a look at some studies on this subject [15–18].

In this study, we investigated some shallow and long wave models, such as the modified Benjamin-Bona-Mahony (MBBM) equation, Ostrovsky-Benjamin-Bona-Mahony (OS-BBM) equation, and the Mikhailov-Novikov-Wang (MNW) equation. Some studies about MBBM can be listed as follows: the traveling wave solution of the model via the modified simple equation method [19], soliton solutions via the bifurcation method [20], and analytic treatment of the model [21]. Also, researchers can find studies about OS-BBM in the literature, such as dark and bright soliton solutions of model [22]. In addition, studies like multi-soliton solutions of MNW [23]

*Corresponding author: Mustafa Bayram, Computer Engineering, Biruni University, Turkey. Email: mustafabayram@biruni.edu.tr

and novel exact solutions of MNW [23] can be listed as recent studies about MNW in the literature.

The organization of the paper is as follows: Section 2 includes some preliminaries, as in Section 2.1, the fundamentals of the PINN method are included, and in Section 2.2, the PINN scheme for the models is included. Section 3 includes implementation of the PINN method on the models. In Section 4, the obtained results of the application are included via tables and figures. Lastly, Section 5 includes a conclusion about the study.

## 2. Preliminaries

In this section, we gave some basic preliminaries about artificial neural networks and the PINN method, and we also gave Python code blocks for the desired equations.

### 2.1. PINN

According to the method, the studied equations can be generalized as follows:

$$\phi t(x,t) + N[\phi(x,t)] = 0, x \in [x0, x1], t \in [t0, t1], \quad (1)$$

where $\phi(x,t)$ is the hidden solution of the PDE and N[.] is the nonlinear operator for $\phi(x,t)$. On the other hand, the net function $f$ can be defined by the left-hand side of Equation (1) as follows:

$$f := \phi t(x,t) + N[\phi(x,t)] \quad (2)$$

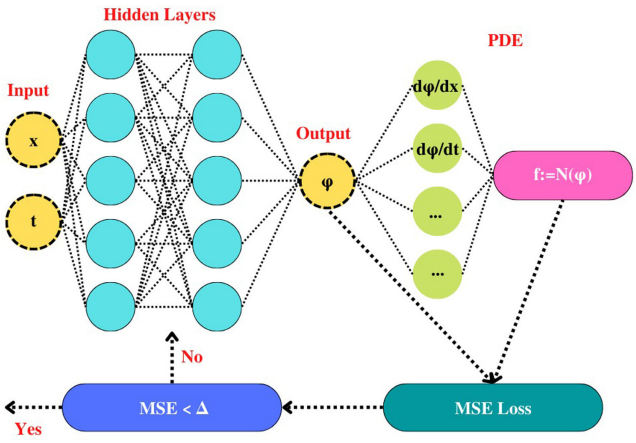
The approximation of $\phi(x,t)$ from $f$ via minimizing mean square error (MSE) and neural networks is called PINN Minimizing MSE loss can be defined as follows:

$$\begin{aligned} MSE &= MSE_\phi + MSE_f, \\ MSE_\phi &= \frac{1}{N_\phi}\sum_{i=1}^{N_\phi}\left|\phi\left(x_\phi^i, t_\phi^i\right) - \phi^i\right|^2, \\ MSE_f &= \frac{1}{N_f}\sum_{i=1}^{N_f}\left|f\left(x_f^i, t_f^i\right)\right|^2, \end{aligned} \quad (3)$$

where $\left\{x_\phi^i, t_\phi^i, \phi^i\right\}_{i=1}^{N_\phi}$ represents initial and boundary data training on $\phi(x,t)$ and $\left\{x_f^i, t_f^i\right\}_{i=1}^{N_f}$ represents collocation points for $f(x,t)$. The $MSE\phi$ loss denotes the loss of initial and boundary values; the MSE $f$ loss denotes the loss of the created structure in Equation (2). In Figure 1, we illustrated the working scheme of the PINN method for solving PDEs. The working principle is as follows: the system is initialized using only the initial values of the PDE. Estimates obtained via initial values are substituted in $f$ in Equation (2), and weight determinations are made by minimizing the error. The system proceeds using the previous prediction value for all points of the grid system.

Also, it should not be forgotten that PINN is a machine learning technique. In the literature, the term black box is used for systems containing artificial neural networks. However, to give a summary for readers to understand more easily, the method predicts the next step in the grid using the initial values and tries to bring the PDE solution closer to zero by substituting the prediction into the $f$ function. This minimization process is done through the loss function. Once all steps in the grid are predicted, the surface solution of a PDE is revealed.

**Figure 1**
**PINN method working scheme**




### 2.2. PINN scheme for MBBM

MBBM equation is given as follows [19];

$$\varphi t + \varphi x - \alpha\varphi 2\varphi x + \varphi xxx = 0 \quad (4)$$

where $\alpha$ is a nonzero constant. The MBBM equation is a model for the surface of long waves in ocean engineering and represents the character of hydromagnetic waves in plasma matter [20, 21]. The real-valued neural network for the PINN scheme can be written as:

**Listing 1**
**Real valued neural network for MBBM equation**

```
def pinn phi (self, x, t):
    phi = self. neural net (tf. concat ([x, t],1),
    self. weights, self. biases) return phi
```

**Listing 2**
**Real valued net function for MBBM equation**

```
def net f (self, x, t):
    phi = self. pinn phi (x, t)
    phi t = tf. gradients (phi, t) [0]
    phi x = tf. gradients (phi, x) [0]
    phi xx = tf. gradients (phi x, x) [0]
    phi xxx = tf. gradients (phi xx, x) [0]
    f = phi t+phi x−alpha * (phi **2) * phi x + phi xxx
    #Equation 1
    return f
```

### 2.3. PINN scheme for OS-BBM

The OS-BBM equation is given as follows [22]:

$$(\phi t + \phi x - \alpha(\phi 2)x - \beta\phi xxt)x - \gamma(\phi + \phi 2) = 0 \quad (5)$$

where $\alpha,\beta,\gamma$ is a nonzero constant. OS-BBM equation models wave motion on the ocean surface and some special cases of magneto-acoustic waves [22]. The real-valued neural network for the PINN scheme can be written as:

**Listing 3**
**Real valued net function for OS-BBM equation**

```
def net f (self, x, t):
    phi = self. pinn phi (x, t)
    phi t = tf. gradients (phi, t) [ 0 ]
    phi x = tf. gradients (phi, x) [ 0 ]
    phi xx = tf. gradients (phi x, x) [ 0 ]
 phi xxt = tf. gradients (phi xx, t) [ 0 ] phi xxx =
 tf. gradients (phi xx, x) [ 0 ]
 f = tf. gradients (phi t + phi x −alpha* tf.
 gradients (phi **2, x) [ 0 ] − beta* phi xxt, x)
 [ 0 ] − gamma (phi + phi **2) #Equation 2
 return f
```

## 2.4. PINN scheme for MNW

The MNW equation is given as follows [23, 24]:

$$\phi tt - \phi xxxt - 8\phi x\phi xt - 4\phi xx\phi t + 2\phi x\phi xxxx + 4\phi xx\phi xxx + 24\phi\phi xx = 0. \tag{6}$$

MNW equations model some special ocean waves and can also be derived from other phenomenon ocean wave models [23–25]. The real-valued neural network for the PINN scheme can be written as:

**Listing 4**
**Real valued net function for MNW equation**

```
def net f (self, x, t):
    phi = self. pinn phi (x, t)
    phi t = tf. gradients (phi, t) [ 0 ]
    phi x = tf. gradients (phi, x) [ 0 ]
    phi xt = tf. gradients (phi x, t) [ 0 ]
    phi xx = tf. gradients (phi x, x) [ 0 ]
    phi tt = tf. gradients (phi t, t) [ 0 ]
    phi xxx = tf. gradients (phi xx, x) [ 0 ]
    phi xxxt= tf. gradients (phi xxx, t) [ 0 ]
    phi xxxx = tf. gradients (phi xxx, x) [ 0 ]
    f = phi tt − phi xxxt − 8 * phi x* phi xt −4* phi xx*
    phi t
 + 2 * phi x* phi xxxx + 4 * u xx * u xxx + 24 * ((phi x)** 2)
 * phi xx return f
```

## 3. Application

In this section, we applied the PINN method to the (1+1)-dimensional PDEs.

## 3.1. MBBM equation

In Khan et al. [19], the kink soliton solution of the Equation (4) is given as:

$$\phi(x,t) = \sqrt{\left(\frac{3(1-\beta)}{\alpha}\right)} \times \tanh\left(\sqrt{\left(\frac{1-\beta}{2}\right)}(x-\beta t)\right) \tag{7}$$

where $\alpha, \beta$ are arbitrary real constants with $\frac{1-\beta}{2} > 0$ and also $\beta$ is wave velocity. To simulate our method, we created a dataset of the exact solution of the Equation (7) via MATLAB. Datasets are created as follows; $x$ and $t$ are generated for 256 equally spaced values in $[-10, 10]$.

## 3.2. OS-BBM equation

In the study by Alquran [22], bright soliton solution of the Equation (5) is given as:

$$\phi(x,t) = \frac{-3}{2}\operatorname{sech}^2\left(\frac{\sqrt{\alpha+\beta\gamma}}{2\sqrt{\beta+\alpha\beta}}\left(x - \left(\frac{\alpha+\alpha^2}{\alpha+\beta\gamma}\right)t\right)\right) \tag{8}$$

where $\alpha, \beta, \gamma$ are real constants, $\left(\frac{\alpha+\alpha^2}{\alpha+\beta\gamma}\right)$ is wave velocity with $\alpha + \beta\gamma > 0$ and $\beta + \alpha\beta > 0$. To simulate our method, we created a dataset as follows; x is generated for 256 equally spaced values in [−10, 10] and t is generated for 256 equally spaced values in [0, 5].

## 3.3. MNW equation

In Akbulut et al. [24], kink type one-soliton solution of the Equation (6) is given as:

$$\phi(x,t) = \frac{2\alpha e^{\alpha^3 t+\alpha x}}{1+e^{\alpha^3 t+\alpha x}} \tag{9}$$

where α is the wavenumber and real constant, and wave speed is $\alpha^3$. The dataset is created via MATLAB with x and t generated for 256 equally spaced values in [−10, 10] for α = 1.2.

In Akbulut et al. [24], kink type two-soliton interaction solution of the Equation (6) is given as:

$$\phi(x,t) = \frac{2\alpha_1 e^{\alpha_1^3 t+\alpha_1 x} + 2\alpha_2 e^{\alpha_2^3 t+\alpha_2 x} + \frac{2(\alpha_1-\alpha_2)^2 e^{(\alpha_1+\alpha_2)x+\left(\alpha_1^3+\alpha_2^3\right)t}}{\alpha_1+\alpha_2}}{1+e^{\alpha_1^3 t+\alpha_1 x} + e^{\alpha_2^3 t+\alpha_2 x} + \frac{(\alpha_1-\alpha_2)^2 e^{(\alpha_1+\alpha_2)x+\left(\alpha_1^3+\alpha_2^3\right)t}}{\alpha_1+\alpha_2}} \tag{10}$$

where $\alpha_1, \alpha_2$ are arbitrary real constants. In this problem, we applied the method to two different parametrizations. Firstly, a dataset is created for $\alpha_1 = -1.41$ and $\alpha_2 = 1.4$ values, and nodes are divided into 256 equally spaced values with $x \in [-20, 20]$ and $t \in [-5, 5]$. Secondly, a dataset is created for $\alpha_1 = -1.5$ and $\alpha_2 = 0.5$ values, and nodes are divided into 256 equally spaced values with $x \in [-20, 20]$ and $t \in [-10, 10]$.

In Saha Ray and Singh [23], mixed dark-bright soliton solution of the Equation (6) is given as:

$$\phi(x,t) = -\frac{x}{3} + \frac{t}{3} - \frac{2}{1+\cosh(-x+t)-\sinh(-x+t)} \tag{11}$$

The dataset is created for 256 equally spaced values with $x, t \in [-5, 5]$. The next chapter describes the results using tables and figures.

## 4. Results and Discussion

In this section, we give the obtained results with tables and figures. In Table 1, a comparison of the obtained results of Equation (7) is given. Results are changing according to initial Nϕ and neural network numbers. According to Table 1, the best results are obtained with 8 hidden layers and 60 neural networks per layer. Absolute error is approximately $1.25 \times 10^{-3}$. In addition, the results are shown in Figure 2. Figure 2a shows a 3D view of the exact solution; Figure 2b shows a 3D view of the approximate solution obtained via the PINN method; Figure 2c

**Table 1**
**Comparison of Equation (7)**

| $N_\phi$ | $Nf$ | Hidden layer | Neural networks | Error u | Time(sec) | Number of iterations |
|---|---|---|---|---|---|---|
| 200 | 4000 | 8 | 20 | $1,729082 \times 10^{-3}$ | 8.89 | 140 |
| 200 | 4000 | 8 | 40 | $1,255546 \times 10^{-3}$ | 39.34 | 197 |
| 200 | 4000 | 8 | 60 | $4,563819 \times 10^{-3}$ | 76.92 | 179 |
| 300 | 4000 | 8 | 100 | $5,315208 \times 10^{-3}$ | 152.20 | 200 |

**Figure 2**
**The various plots of $\phi$(x,t) in Equation (7)**

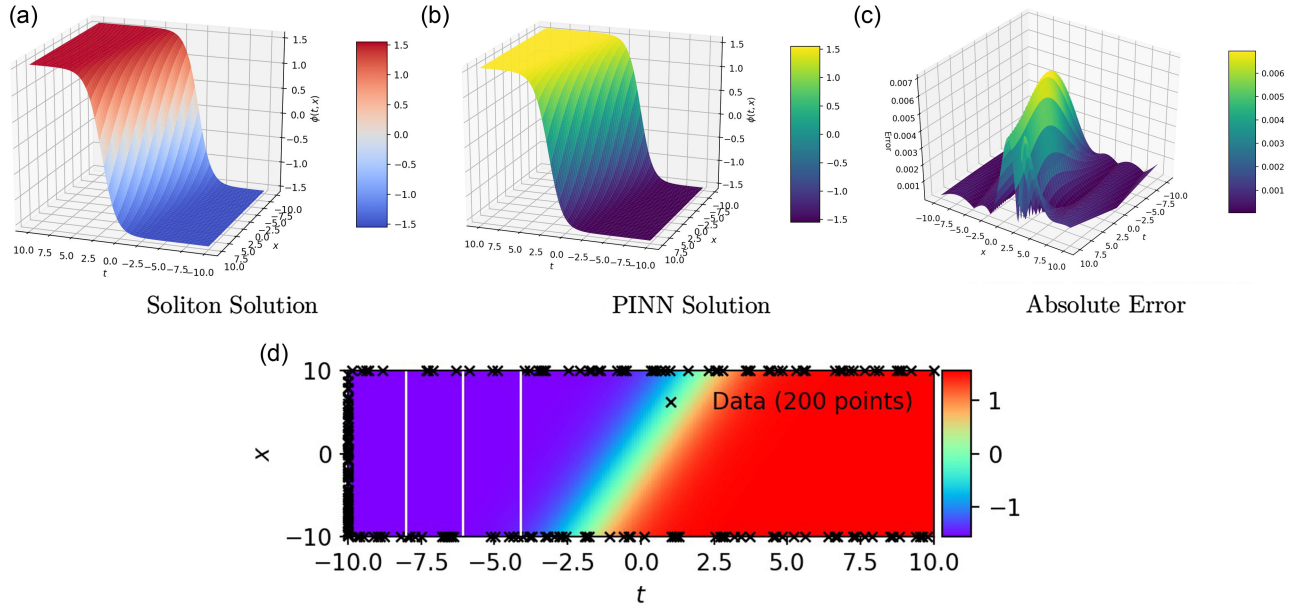



**Table 2**
**Comparison of Equation (8)**

| $N_\phi$ | $Nf$ | Hidden layer | Neural networks | Error u | Time(sec) | Number of iterations |
|---|---|---|---|---|---|---|
| 200 | 4000 | 8 | 20 | $8,02994 \times 10^{-1}$ | 55.92 | 105 |
| 200 | 4000 | 8 | 40 | $4,284380 \times 10^{-1}$ | 1499.21 | 371 |
| 200 | 4000 | 8 | 60 | $7,817779 \times 10^{-1}$ | 776.05 | 151 |
| 300 | 4000 | 8 | 100 | $1,825055 \times 10^{-3}$ | 1197.72 | 278 |

shows absolute error; and Figure 2d shows a contour plot of the approximate solution.

Table 2 shows the comparison of obtained results for Equation (8). According to Table 2, the best results are obtained via 8 hidden layers with 100 neural networks per layer. Absolute error is approximately $1.82 \times 10^{-3}$. Figure 3 shows the results for Equation (8). In Figure 3a, the soliton solution of Equation (8) is shown. Figure 3b shows the PINN solution, and Figure 3c shows the absolute error between soliton and the PINN solution. In Figure 3d, a contour plot of the approximate solution obtained via the PINN method is depicted.

According to Table 3, the best results are obtained with 8 hidden layers and 40 neural networks per layer. The absolute error is approximately $2.75 \times 10^{-2}$. In addition, the results are shown in Figure 4. Figure 4a shows a 3D view of the Soliton solution; Figure 4b shows a 3D view of the approximate solution obtained via the PINN method; Figure 4c shows the absolute error; and Figure 4d shows the contour plot of the PINN solution.

Table 4 shows the comparison of obtained results for Equation (10). According to Table 4, the best results are obtained via 8 hidden layers with 20 neural networks per layer. Absolute error is approximately $2.20 \times 10^{-2}$. Figure 5 shows the results for Equation (10). In Figure 5a, the soliton solution of Equation (10) is shown. Figure 5b shows the PINN solution, and Figure 5c shows the absolute error between the soliton solution and the PINN solution. In Figure 5d, a contour plot of the approximate solution obtained via the PINN method is depicted.

**Figure 3**
**The various plots of $\phi$(x,t) in Equation (8)**

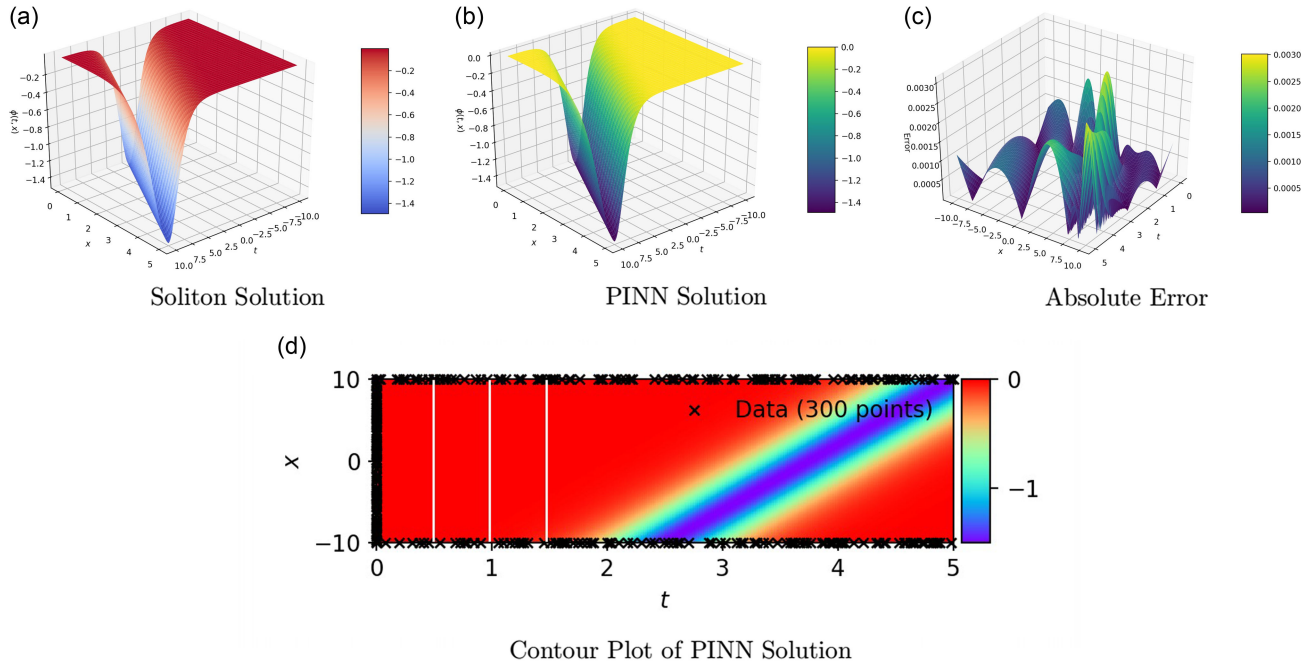



**Table 3**
**Comparison of Equation (9)**

| $N_\phi$ | $Nf$ | Hidden layer | Neural networks | Error u | Time(sec) | Number of iterations |
|---|---|---|---|---|---|---|
| 200 | 4000 | 8 | 20 | $3,797476 \times 10^{-2}$ | 47.65 | 133 |
| 200 | 4000 | 8 | 40 | $2,750077 \times 10^{-2}$ | 91.62 | 78 |
| 200 | 4000 | 8 | 60 | $5,433307 \times 10^{-2}$ | 165.48 | 70 |
| 300 | 4000 | 8 | 100 | $6,953371 \times 10^{-2}$ | 293.65 | 65 |

**Figure 4**
**The various plots of $\phi$(x,t) in Equation (9)**

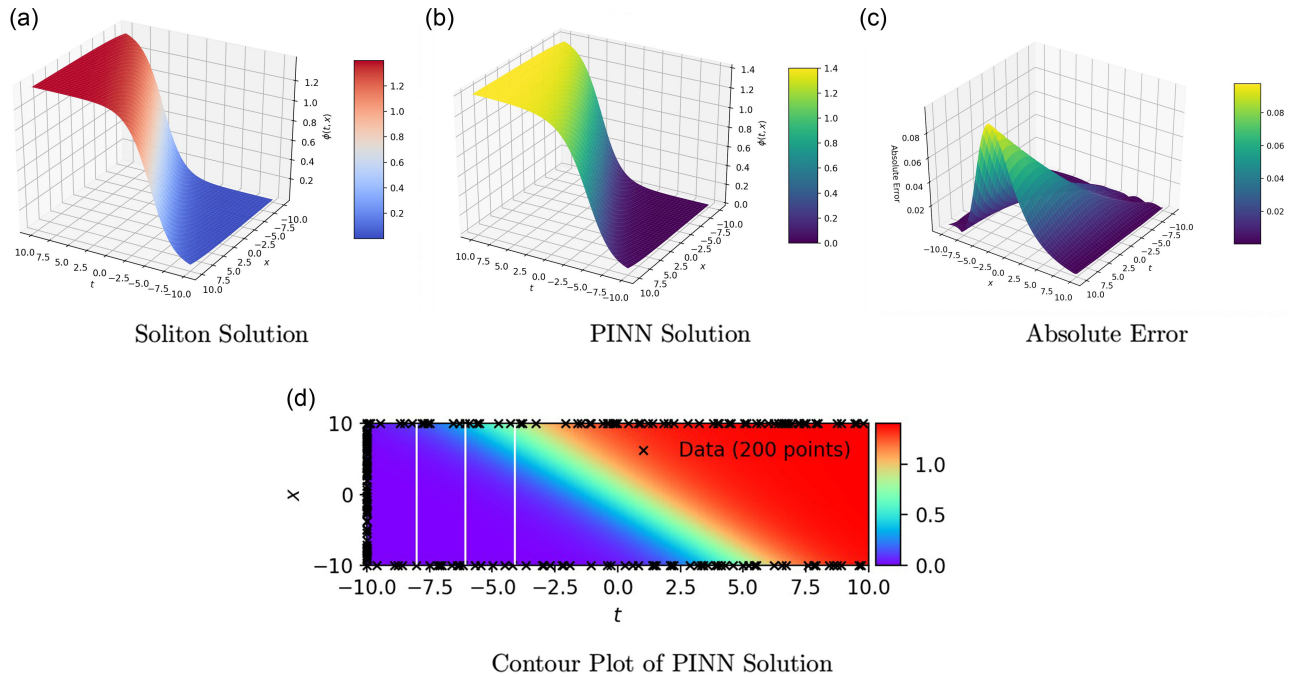


**Table 4**
**Comparison of Equation (10)**

| $N_\phi$ | $Nf$ | Hidden layer | Neural networks | Error u | Time(sec) | Number of iterations |
|---|---|---|---|---|---|---|
| 200 | 4000 | 8 | 20 | $2,200254 \times 10^{-2}$ | 1536.63 | 6327 |
| 200 | 4000 | 8 | 40 | $2,697991 \times 10^{-1}$ | 5683.23 | 5592 |
| 200 | 4000 | 8 | 60 | $2,405252 \times 10^{-1}$ | 67103.70 | 7912 |
| 300 | 4000 | 8 | 100 | $2,766252 \times 10^{-1}$ | 180084.39 | 4374 |

**Figure 5**
**The various plots of $\phi$(x,t) in Equation (10)**

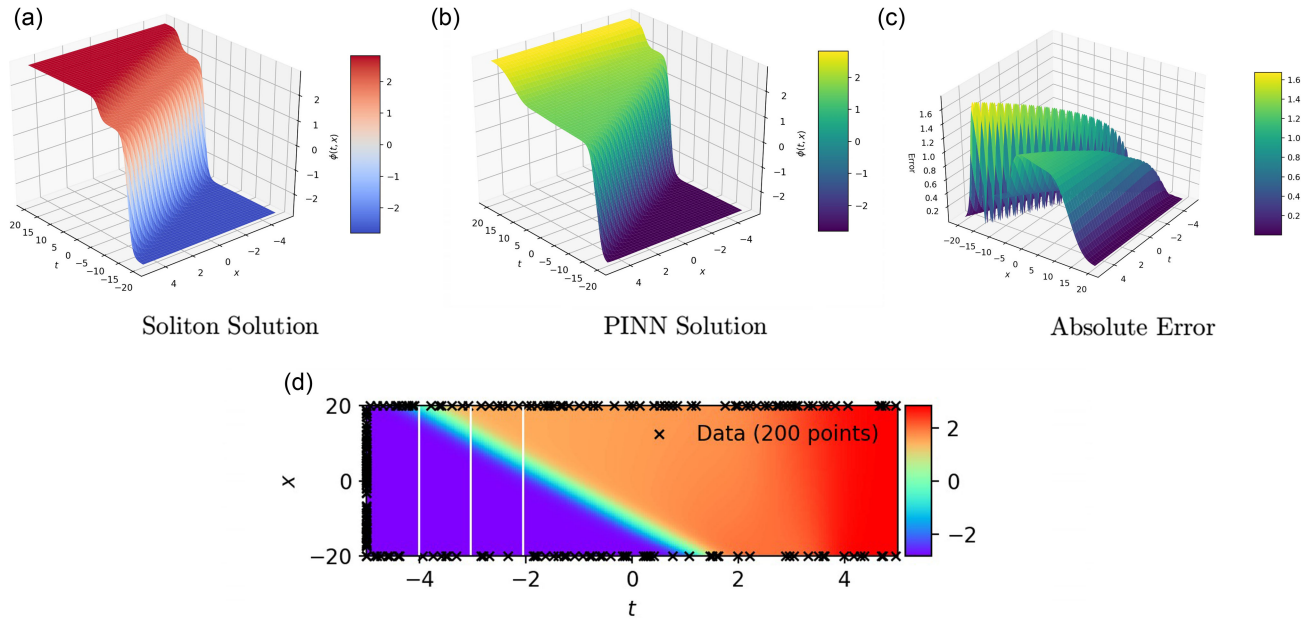



**Table 5**
**Comparison of Equation (10)**

| $N_\phi$ | $Nf$ | Hidden layer | Neural networks | Error u | Time(sec) | Number of iterations |
|---|---|---|---|---|---|---|
| 200 | 4000 | 8 | 20 | $4,969024 \times 10^{-1}$ | 944.8 | 3796 |
| 200 | 4000 | 8 | 40 | $5,042329 \times 10^{-1}$ | 7480.73 | 3485 |
| 200 | 4000 | 8 | 60 | $4,944566 \times 10^{-1}$ | 8487.50 | 2805 |
| 300 | 4000 | 8 | 100 | $5,174641 \times 10^{-1}$ | 78819.09 | 2970 |

According to Table 5, the best results are obtained with 8 hidden layers and 60 neural networks per layer. The absolute error is approximately $4.94 \times 10^{-1}$. In addition, the results are shown in Figure 6. Figure 6a shows a 3D view of the soliton solution; Figure 6b shows a 3D view of the approximate solution obtained via the PINN method; Figure 6c shows the absolute error; and Figure 6d shows the contour plot of the PINN solution.

Table 6 shows the comparison of obtained results for Equation (11). According to Table 6, the best results are obtained via 8 hidden layers with 20 neural networks per layer. The absolute error is approximately $7.05 \times 10^{-4}$. Figure 7 shows the results for Equation (11). In Figure 7a, the soliton solution of Equation (11) is shown. Figure 7b shows the PINN solution, and Figure 7c shows the absolute error between the soliton solution and the PINN solution. In Figure 7d, a contour plot of the approximate solution obtained via the PINN method is depicted.

When evaluating the results obtained in this study, we can start with the scope. When looking at the studies on obtaining solitons using PINN [4, 26, 27], it can be seen that the approximate error rate in existing studies is achieved. While other studies have reached an error rate of approximately $10^{-3}$, these values are also present in our study. Another thing we need to compare is the types of solitons in existing studies. Looking at the study examining the two-soliton, rogue structure [26, 28, 29] which are the rare types in the literature, it will be seen that the error obtained is similar to the error in this study. It can be seen that the

**Figure 6**
**The various plots of $\phi$(x,t) in Equation (10)**

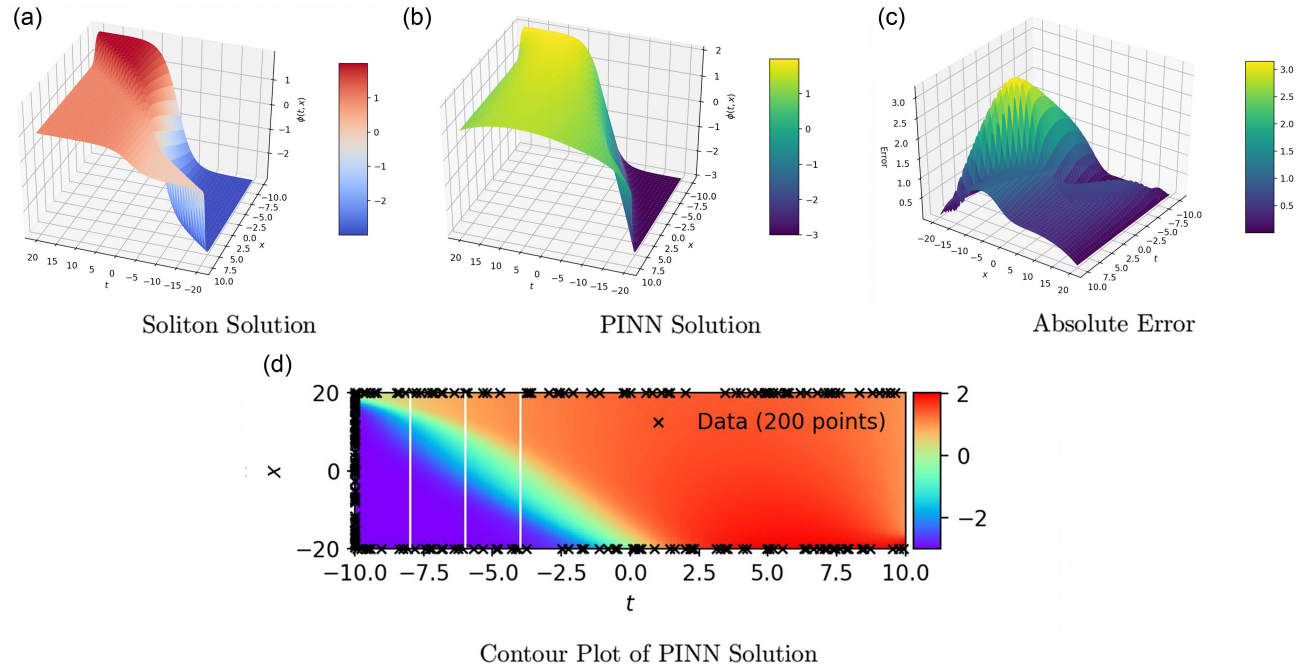



**Table 6**
**Comparison of Equation (11)**

| $N_\phi$ | $Nf$ | Hidden layer | Neural networks | Error u | Time(sec) | Number of iterations |
|---|---|---|---|---|---|---|
| 200 | 4000 | 8 | 20 | $7,055715 \times 10^{-4}$ | 89.6 | 300 |
| 200 | 4000 | 8 | 40 | $1,80200 \times 10^{-3}$ | 2697.15 | 703 |
| 200 | 4000 | 8 | 60 | $6,029168 \times 10^{-3}$ | 1640.96 | 545 |
| 300 | 4000 | 8 | 100 | $4,308377 \times 10^{-3}$ | 1307.51 | 369 |

classical PINN method gives poor results compared to advanced numerical methods such as B-Spline and the modified Laplace decomposition method in layered and complex soliton types [30, 31].

Another unpredictable weakness of the PINN method is that it does not have a fixed network structure. Examining some other PINN studies revealed different error rates for varying numbers of hidden layers. Some of these studies and their results are shown in Table 7 for easier viewing.

As can be seen in Table 7, there is no correlation between the increase or decrease in the number of hidden layers or neurons and the success of the method. Therefore, researchers will have to try many times to find the most accurate result. A disadvantage can also be seen in this aspect. Another conclusion that can be drawn from the table is that the PINN method can be applied to equation types in many different fields. Although only 1D water wave models were considered in this study, other ocean engineering examples where the PINN method is used can be listed as follows: Examining the shallow water equation and advection-diffusion equation with PINN and their effects on climate change [36] and ocean temperature [37], examination of local advection-diffusion equations with the PINN method and uncertainty in ocean modeling [38], and examination of internal soliton movements modeled by the Kadomtsev-Petviashvili equation with PINN and comparison of the results with satellite images [39]. In addition, when looking at the time-dependent calculation cost in Table 7, a definitive conclusion cannot be made, but there are methods in the literature that reduce the time cost, such as CAN-PINN [11]. On the other hand, another point that concerns the success of the method is the initial and boundary conditions. Some customized PINN models [40–43] have included these conditions in the loss function and achieved more successful results.

Since the PINN method is a machine learning method, we can mention some hyperparameters that affect its success. These can be listed as learning rate, activation function, optimization algorithm, number of hidden layers, and neurons. This study solely utilized tables to illustrate the impact of altering the number of hidden layers and neurons on success. However, there are currently studies showing the effect of other hyperparameters on the success of PINN. Some of these studies are as follows: the effect of hyperparameters on error convergence [44], the effect of parameters such as activation function and network architecture on the error rate [45], and some other similar studies [46–48].

## 5. Conclusion

In this work, we have studied approximate solutions of one-dimensional shallow water waves such as the MBBM, OS-BBM, and MNW equations. Based on this work, we have identified several soliton structures of water waves, including kink, dark, kink with two solitons interacting, and mixed bright-dark solitons.

**Figure 7**
**The various plots of $\phi$(x,t) in Equation (11)**

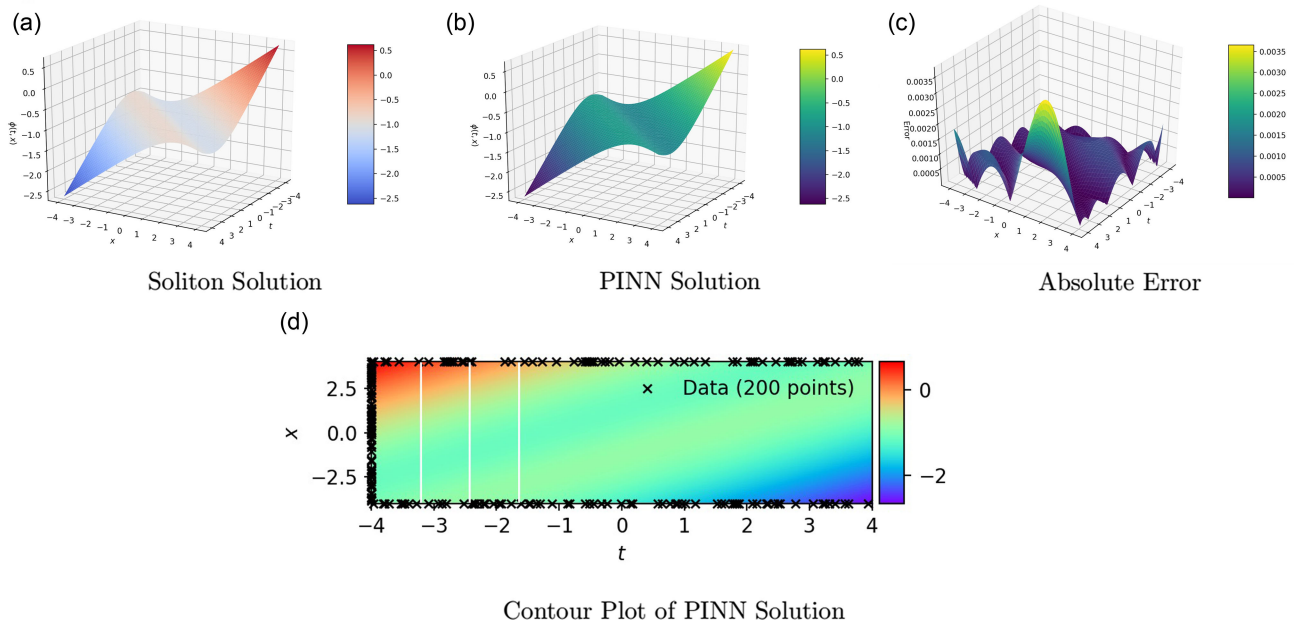



**Table 7**
**Comparison of results**

| Author | Equation | Setup | Error | Time(sec) |
|---|---|---|---|---|
| Fang et al. [4] | High order nonlinear Schrödinger (NLSE) | 7 hidden layers × 30 neurons | $9.1252 \times 10^{-3}$ | 5545.7 |
| Cui et al. [32] | Boussinesq | 4 hidden layers × 40 neurons | $2.0 \times 10^{-2}$ | 152 |
| Wang et al. [33] | Generalized Hirota-Satsuma coupled KdV | 5 hidden layers × 20 neurons | $6.949 \times 10^{-5}$ | 607.9 |
| Peng and Chen [34] | Nonlocal Hirota | 9 hidden layers × 40 neurons | $2.695 \times 10^{-4}$ | 1262.4 |
| Yuan et al. [35] | Volterra | 4 hidden layers × 80 neurons | $9.17 \times 10^{-5}$ | N/A |

On the other hand, this is the first study about the mentioned models using the PINN method. Moreover, we have reached the absolute error in the $10^{-1}$ to $10^{-4}$ interval. We understand that some of the error values obtained in our study are relatively high compared to other studies in the literature. However, some of the error values we obtained in our study also yielded reasonably good and effective results. Examining soliton types with high error rates leads to the conclusion that the complex and layered soliton structures prevent numerical solutions from yielding satisfactory results. Future researchers can set the goal of achieving better solutions in layered solitons. The solution of models used in fields such as water waves and ocean engineering, which include one-dimensional equations, as well as in the field of optics, which includes larger-dimensional and complex equations, may be one of the future goals. Another future target can be set as working on higher-dimensional or complex-valued models that are used in the optic field. Some studies have stated that they have run the system successfully and obtained results, especially for high-dimensional equations in the field of optics. However, these studies are quite incomplete compared to traditional methods in terms of absolute error. Therefore, the issue of reducing the error rate is one of the important expectations for future studies.

## Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

## Data Availability Statement

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

## Author Contribution Statement

**Ismail Onder:** Conceptualization, Methodology, Software, Formal analysis. **Mustafa Bayram:** Validation, Investigation, Resources, Data curation. **Abdulkadir Sahiner:** Writing – original draft, Project administration. **Aydin Secer:** Writing – review & editing, Visualization, Supervision.

## References

[1] Bakthavatchalam, T. A., Ramamoorthy, S., Sankarasubbu, M., Ramaswamy, R., & Sethuraman, V. (2021). Bayesian optimization of Bose-Einstein condensates. *Scientific Reports*, *11*(1), 5054. https://doi.org/10.1038/s41598-021-84336-0

[2] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

[3] Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems

involving nonlinear partial differential equations. *Journal of Computational Physics*, *378*, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045

[4] Fang, Y., Wu, G. Z., Wang, Y. Y., & Dai, C. Q. (2021). Data-driven femtosecond optical soliton excitations and parameters discovery of the high-order NLSE using the PINN. *Nonlinear Dynamics*, *105*(1), 603–616. https://doi.org/10.1007/s11071-021-06550-9

[5] Li, J., & Chen, Y. (2020). A deep learning method for solving third-order nonlinear evolution equations. *Communications in Theoretical Physics*, *72*(11), 115003. https://doi.org/10.1088/1572-9494/abb7c8

[6] Wu, G. Z., Fang, Y., Kudryashov, N. A., Wang, Y. Y., & Dai, C. Q. (2022). Prediction of optical solitons using an improved physics-informed neural network method with the conservation law constraint. *Chaos, Solitons & Fractals*, *159*, 112143. https://doi.org/10.1016/j.chaos.2022.112143

[7] Peng, W. Q., Pu, J. C., & Chen, Y. (2022). PINN deep learning method for the Chen–Lee–Liu equation: Rogue wave on the periodic background. *Communications in Nonlinear Science and Numerical Simulation*, *105*, 106067. https://doi.org/10.1016/j.cnsns.2021.106067

[8] Lin, S., & Chen, Y. (2022). A two-stage physics-informed neural network method based on conserved quantities and applications in localized wave solutions. *Journal of Computational Physics*, *457*, 111053. https://doi.org/10.1016/j.jcp.2022.111053

[9] Wen, X. K., Wu, G. Z., Liu, W., & Dai, C. Q. (2022). Dynamics of diverse data-driven solitons for the three-component coupled nonlinear Schrödinger model by the MPS-PINN method. *Nonlinear Dynamics*, *109*(4), 3041–3050. https://doi.org/10.1007/S11071-022-07583-4

[10] Pu, J., Li, J., & Chen, Y. (2021). Solving localized wave solutions of the derivative nonlinear Schrödinger equation using an improved PINN method. *Nonlinear Dynamics*, *105*, 1723–1739. https://doi.org/10.1007/S11071-021-06554-5

[11] Chiu, P. H., Wong, J. C., Ooi, C., Dao, M. H., & Ong, Y. S. (2022). CAN-PINN: A fast physics-informed neural network based on coupled-automatic–numerical differentiation method. *Computer Methods in Applied Mechanics and Engineering*, *395*, 114909. https://doi.org/10.1016/J.CMA.2022.114909

[12] Fang, Y., Wu, G. Z., Wen, X. K., Wang, Y. Y., & Dai, C. Q. (2022). Predicting certain vector optical solitons via the conservation-law deep-learning method. *Optics & Laser Technology*, *155*, 108428. https://doi.org/10.1016/J.OPTLASTEC.2022.108428

[13] Esen, H., Ozdemir, N., Secer, A., & Bayram, M. (2023). Traveling wave structures of some fourth-order nonlinear partial differential equations. *Journal of Ocean Engineering and Science*, *8*(2), 124–132. https://doi.org/10.1016/j.joes.2021.12.006

[14] Onder, I., Cinar, M., Secer, A., & Bayram, M. (2024). Analytical solutions of simplified modified Camassa-Holm equation with conformable and M-truncated derivatives: A comparative study. *Journal of Ocean Engineering and Science*, *9*(3), 240–250. https://doi.org/10.1016/j.joes.2022.06.012

[15] Zhang, R., & Gao, Y. (2024). Learning scattering waves via coupling physics-informed neural networks and their convergence analysis. *Journal of Computational and Applied Mathematics*, *446*, 115874. https://doi.org/10.1016/j.cam.2024.115874

[16] Meng, Y., Zhou, R., Mukherjee, A., Fitzsimmons, M., Song, C., & Liu, J. (2024). Physics-informed neural network policy iteration: Algorithms, convergence, and verification. *arXiv Preprint:2402.10119*.

[17] Shin, Y., Darbon, J., & Karniadakis, G. E. (2020). On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs. *arXiv Preprint:2004.01806*.

[18] Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., & Piccialli, F. (2022). Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, *92*(3), 88. https://doi.org/10.1007/s10915-022-01939-z

[19] Khan, K., Akbar, M. A., & Islam, S. M. R. (2014). Exact solutions for (1+1)-dimensional nonlinear dispersive modified Benjamin-Bona-Mahony equation and coupled Klein-Gordon equations. *SpringerPlus*, *3*, 724. https://doi.org/10.1186/2193-1801-3-724

[20] Song, M. (2015). Nonlinear wave solutions and their relations for the modified Benjamin–Bona–Mahony equation. *Nonlinear Dynamics*, *80*(1), 431–446. https://doi.org/10.1007/s11071-014-1880-5

[21] Abbasbandy, S., & Shirzadi, A. (2010). The first integral method for modified Benjamin–Bona–Mahony equation. *Communications in Nonlinear Science and Numerical Simulation*, *15*(7), 1759–1764. https://doi.org/10.1016/j.cnsns.2009.08.003

[22] Alquran, M. (2012). Bright and dark soliton solutions to the Ostrovsky-Benjamin-Bona-Mahony (OS-BBM) equation. *Journal of Mathematical and Computational Science*, *2*(1), 15–22.

[23] Saha Ray, S., & Singh, S. (2021). New various multisoliton kink-type solutions of the (1+1)-dimensional Mikhailov–Novikov–Wang equation. *Mathematical Methods in the Applied Sciences*, *44*(18), 14690–14702. https://doi.org/10.1002/mma.7736

[24] Akbulut, A., Kaplan, M., & Kaabar, M. K. A. (2023). New exact solutions of the Mikhailov-Novikov-Wang equation via three novel techniques. *Journal of Ocean Engineering and Science*, *8*(1), 103–110. https://doi.org/10.1016/j.joes.2021.12.004

[25] Mikhailov, A. V., Novikov, V. S., & Wang, J. P. (2007). On classification of integrable nonevolutionary equations. *Studies in Applied Mathematics*, *118*(4), 419–457. https://doi.org/10.1111/j.1467-9590.2007.00376.x

[26] Fang, Y., Wu, G. Z., Kudryashov, N. A., Wang, Y. Y., & Dai, C. Q. (2022). Data-driven soliton solutions and model parameters of nonlinear wave models via the conservation-law constrained neural network method. *Chaos, Solitons & Fractals*, *158*, 112118. https://doi.org/10.1016/j.chaos.2022.112118

[27] Jiang, X., Wang, D., Fan, Q., Zhang, M., Lu, C., & Lau, A. P. T. (2022). Physics-informed neural network for nonlinear dynamics in fiber optics. *Laser & Photonics Reviews*, *16*(9), 2100483. https://doi.org/10.1002/lpor.202100483

[28] Fang, Y., Bo, W. B., Wang, R. R., Wang, Y. Y., & Dai, C. Q. (2022). Predicting nonlinear dynamics of optical solitons in optical fiber via the SCPINN. *Chaos, Solitons & Fractals*, *165*, 112908. https://doi.org/10.1016/j.chaos.2022.112908

[29] Wu, G. Z., Fang, Y., Wang, Y. Y., Wu, G. C., & Dai, C. Q. (2021). Predicting the dynamic process and model parameters of the vector optical solitons in birefringent fibers

via the modified PINN. *Chaos, Solitons & Fractals*, *152*, 111393. https://doi.org/10.1016/j.chaos.2021.111393

[30] Başhan, A., & Esen, A. (2021). Single soliton and double soliton solutions of the quadratic-nonlinear Korteweg-de Vries equation for small and long-times. *Numerical Methods for Partial Differential Equations*, *37*(2), 1561–1582. https://doi.org/10.1002/num.22597

[31] Cinar, M., Onder, I., Secer, A., Bayram, M., Abdulkadir Sulaiman, T., & Yusuf, A. (2022). Solving the fractional Jaulent–Miodek system via a modified Laplace decomposition method. *Waves in Random and Complex Media*, 1–14. https://doi.org/10.1080/17455030.2022.2057613

[32] Cui, S., Wang, Z., Han, J., Cui, X., & Meng, Q. (2022). A deep learning method for solving high-order nonlinear soliton equations. *Communications in Theoretical Physics*, *74*(7), 075007. https://doi.org/10.1088/1572-9494/ac7202

[33] Wang, X., Wu, Z., Song, J., Han, W., & Yan, Z. (2024). Data-driven soliton solutions and parameters discovery of the coupled nonlinear wave equations via a deep learning method. *Chaos, Solitons & Fractals*, *180*, 114509. https://doi.org/10.1016/j.chaos.2024.114509

[34] Peng, W. Q., & Chen, Y. (2022). N-double poles solutions for nonlocal Hirota equation with nonzero boundary conditions using Riemann–Hilbert method and PINN algorithm. *Physica D: Nonlinear Phenomena*, *435*, 133274. https://doi.org/10.1016/j.physd.2022.133274

[35] Yuan, L., Ni, Y. Q., Deng, X. Y., & Hao, S. (2022). A-PINN: Auxiliary physics informed neural networks for forward and inverse problems of nonlinear integro-differential equations. *Journal of Computational Physics*, *462*, 111260. https://doi.org/10.1016/j.jcp.2022.111260

[36] de Wolff, T., Carrillo, H., Martí, L., & Sanchez-Pi, N. (2021). Assessing physics informed neural networks in ocean modelling and climate change applications. In *AI: Modeling Oceans and Climate Change Workshop at ICLR 2021*.

[37] de Wolff, T., Carrillo, H., Martí, L., & Sanchez-Pi, N. (2021). Towards optimally weighted physics-informed neural networks in ocean modelling. *arXiv Preprint:2106.08747*.

[38] Lütjens, B., Crawford, C. H., Veillette, M., & Newman, D. (2021). PCE-PINNs: Physics-informed neural networks for uncertainty propagation in ocean modeling. *arXiv Preprint:2105.02939*.

[39] Sun, J., Chen, Y., & Tang, X. (2024). Physics-informed neural networks with two weighted loss function methods for interactions of two-dimensional oceanic internal solitary waves. *Journal of Systems Science and Complexity*, *37*(2), 545–566. https://doi.org/10.1007/s11424-024-3500-x

[40] Sukumar, N., & Srivastava, A. (2022). Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. *Computer Methods in Applied Mechanics and Engineering*, *389*, 114333. https://doi.org/10.1016/j.cma.2021.114333

[41] Alkhadhr, S., & Almekkawy, M. (2023). Wave equation modeling via physics-informed neural networks: Models of soft and hard constraints for initial and boundary conditions. *Sensors*, *23*(5), 2792. https://doi.org/10.3390/s23052792

[42] Xie, Y., Ma, Y., & Wang, Y. (2023). Automatic boundary fitting framework of boundary dependent physics-informed neural network solving partial differential equation with complex boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, *414*, 116139. https://doi.org/10.1016/j.cma.2023.116139

[43] Huang, Y. H., Xu, Z., Qian, C., & Liu, L. (2023). Solving free-surface problems for non-shallow water using boundary and initial conditions-free physics-informed neural network (bif-PINN). *Journal of Computational Physics*, *479*, 112003. https://doi.org/10.1016/j.jcp.2023.112003

[44] Pantidis, P., Eldababy, H., Tagle, C. M., & Mobasher, M. E. (2023). Error convergence and engineering-guided hyperparameter search of PINNs: Towards optimized I-FENN performance. *Computer Methods in Applied Mechanics and Engineering*, *414*, 116160. https://doi.org/10.1016/j.cma.2023.116160

[45] Sharma, P., Evans, L., Tindall, M., & Nithiarasu, P. (2023). Hyperparameter selection for physics-informed neural networks (PINNs) – Application to discontinuous heat conduction problems. *Numerical Heat Transfer, Part B: Fundamentals*, 1–15. https://doi.org/10.1080/10407790.2023.2264489

[46] Penwarden, M., Zhe, S., Narayan, A., & Kirby, R. M. (2023). A metalearning approach for physics-informed neural networks (PINNs): Application to parameterized PDEs. *Journal of Computational Physics*, *477*, 111912. https://doi.org/10.1016/j.jcp.2023.111912

[47] Meng, Z., Qian, Q., Xu, M., Yu, B., Yıldız, A. R., & Mirjalili, S. (2023). PINN-FORM: A new physics-informed neural network for reliability analysis with partial differential equation. *Computer Methods in Applied Mechanics and Engineering*, *414*, 116172. https://doi.org/10.1016/j.cma.2023.116172

[48] Polyakov, D. N., & Stepanova, M. M. (2023). Hyperparameter tuning of neural network for high-dimensional problems in the case of Helmholtz equation. *Moscow University Physics Bulletin*, *78*, S243–S255. https://doi.org/10.3103/S0027134923070263

**How to Cite:** Onder, I., Sahiner, A., Secer, A., & Bayram, M. (2024). Soliton Solutions of Some Ocean Waves Supported by Physics Informed Neural Network Method. *Artificial Intelligence and Applications*, *2*(4), 271–280. https://doi.org/10.47852/bonviewAIA42022277