mathematics

MDPI

Article

Prediction of Projectile Interception Point and Interception Time Based on Harris Hawk Optimization–Convolutional Neural Network–Support Vector Regression Algorithm

Zhanpeng Gao and Wenjun Yi *

National Key Laboratory of Transient Physics, Nanjing University of Science and Technology, Nanjing 210094, China; gaozhanpeng@njust.edu.cn
* Correspondences: wenjunyi@njust.edu.cn

Abstract: In modern warfare, the accurate prediction of the intercept time and intercept point of the interceptor is the key to achieving penetration. Aiming at this problem, firstly, a convolutional neural network (CNN) is used to automatically extract high-level features from the data, and then these features are used as the input of support vector regression (SVR) for regression prediction. The Harris Hawk optimization (HHO) is used to optimize the hyperparameters of SVR, and the HHO-CNN-SVR algorithm is proposed. In order to verify the effectiveness of the algorithm for the prediction of the interception point and interception time, this paper constructs a dataset to test the method of simulating the missile interception maneuvering target. Compared with BP, ELM, SVR, HHO-SVR, and CNN-SVR models, the HHO-CNN-SVR model has outstanding performance in prediction accuracy and stability, especially for the interception time. The error is the smallest, and the error fluctuation is small. The MAE of the prediction result is only 0.0139 s; in the interception point prediction, the error of the range and elevation direction is significantly lower than that of the models used for comparison. The MAE in the range direction is 2.3 m, and the MAE in the elevation direction is 2.01 m, which meet the engineering requirements. The HHO-CNN-SVR model has strong prediction accuracy and stability in interception time and interception point prediction. In addition, different control strategies are used to construct a new prediction set, and noise is added to the prediction set. The HHO-CNN-SVR algorithm can maintain good prediction results. The results show that the HHO-CNN-SVR model proposed in this paper has strong generalization ability and high robustness, which can provide reliable support for penetration decision making and defense system optimization.

Keywords: convolutional neural network; support vector regression; Harris Hawk optimization; interception point prediction; penetration

MSC: 68T07

check for updates

Academic Editor: Theodore E. Simos

Received: 5 January 2025
Revised: 16 January 2025
Accepted: 19 January 2025
Published: 21 January 2025

Citation: Gao, Z.; Yi, W. Prediction of Projectile Interception Point and Interception Time Based on Harris Hawk Optimization–Convolutional Neural Network–Support Vector Regression Algorithm. *Mathematics* **2025**, *13*, 338. https://doi.org/10.3390/math13030338

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In modern warfare, the game between offensive missiles (such as cruise missiles, ballistic missiles) and air defense systems is becoming increasingly fierce [1–4]. The enemy's air defense system tries to intercept offensive missiles at the right time and position by accurately predicting the trajectory and flight path of the offensive missiles [5].

In order to effectively break through the interception of enemy air defense systems, our offensive missiles must avoid these interception points and interception time through

precise strategy design. The prediction of the enemy's interception point and interception time determines whether our offensive missiles can successfully penetrate the defense.

The interception point means that the enemy's air defense system calculates the spatial position where the interceptor may be encountered according to the flight path and speed of the offensive missile [6]; the interception time refers to the specific moment when the enemy interceptor encounters our offensive projectile [7]. These predictions not only are critical to the enemy's interception capabilities but also affect our offensive missile penetration strategy. If we can accurately predict the enemy's interception point and interception time, we can increase the probability of successful penetration by evading these points and time using nonlinear trajectories, anti-reconnaissance measures, or decoys.

In order to accurately predict the enemy's interception point and interception time, there are three commonly used prediction methods: kinematic model prediction, physical model prediction, and prediction based on machine learning.

The kinematics model calculates the trajectory by analyzing basic parameters such as the initial velocity and flight direction of the attack projectile and the interceptor, so as to predict the possible interception point and interception time of the enemy interceptor [8,9]. However, the shortcomings of the kinematic model are obvious: it assumes that the flight process is not affected by complex environmental factors, ignoring the actual impact of air resistance, meteorological changes, etc., on the missile flight trajectory and often fails to provide accurate interception points and interception time.

The physical model prediction method is relatively accurate. It considers factors such as air resistance, gravity, wind speed, and temperature by simulating the dynamic behavior of the attack projectile and the interceptor. This method overcomes the limitations of the kinematic model to a certain extent and can better reflect the actual flight state of the attack projectile and the interceptor. Due to the unknown parameters of the enemy interceptor, the kinematic modeling and physical modeling prediction methods are used to break away from the real combat scene. The prediction results are extremely unstable, and the physical model may still have large prediction errors. Therefore, they are not usually used for the interception point and interception time prediction task.

However, in the modern war environment, the combat cycle is generally long, and the application of guided projectiles is increasingly widespread. Because these weapons are frequently launched in combat, they can provide a large amount of actual combat data for subsequent tactical analysis and decision making. Through a large amount of training using historical data, a machine learning model can identify the movement pattern of an attack projectile and learn effective prediction strategies from it [10–12].

Compared with traditional statistical regression methods, machine learning can automatically capture the nonlinear relationships and complex patterns in real-world data. It does not depend on prior assumptions and can handle high-dimensional and multivariate features [13]. With a large amount of actual combat data, high prediction accuracy can be achieved; even in the face of noisy data or data with strong nonlinear trends [14,15], effective prediction results can still be provided. In addition, as the amount of data increases, the machine learning model can be continuously updated and improved, thereby improving the accuracy and stability of the prediction.

The method of establishing a mathematical model between complex data mainly involves the use of a regression prediction algorithm [16]. Lin et al. [17] integrated the advantages of support vector machine and the K-nearest neighbor method and applied the combined model to traffic flow prediction. Jasmin et al. [18] used the LASSO algorithm to predict crop yield. The results showed that the LASSO algorithm can effectively identify the input characteristics related to crop yield and make modeling predictions. Lin et al. [19] improved the whale optimization algorithm and combined it with a GRU

model to achieve good results in the prediction task, which greatly improved the prediction ability of the model.

The selection of hyperparameters of machine learning algorithms directly affects the performance and generalization ability of the model. Traditional hyperparameter selection methods, such as manual debugging or grid search, are often time-consuming and computationally expensive, while intelligent optimization algorithms (such as genetic algorithms [20], particle swarm optimization [21], and Bayesian optimization [22]) can find better hyperparameter combinations in a shorter time. Hamed Khajavi et al. [23] used intelligent optimization algorithms to optimize the hyperparameters of support vector machines, which effectively improved the ability of support vector machines to process nonlinear data and accurately predict energy consumption. Hua et al. [24] improved the atomic search algorithm and used it to optimize an extreme learning machine to predict wind speed. Its superior prediction results also improved the utilization efficiency of wind. Intelligent optimization algorithms have been widely used to optimize machine learning and have achieved good results [25–27].

Therefore, this study used the HHO algorithm [28] to automatically adjust the hyperparameters of SVR, so that it could efficiently explore the parameter space, avoid local optimal problems, and improve model performance and generalization ability. Due to the limitations of traditional SVR in feature selection, CNNs can effectively capture the features in the data that are beneficial to the regression problem through the convolution layer and the pooling layer. Therefore, this study used a CNN to automatically extract the high-level features from the data and then uses these features as the input of SVR for regression prediction [29], thus developing the HHO-CNN-SVR interception point and interception time prediction model.

The main contributions of this paper are as follows:

1. Considering the combat environment and combat cycle, we make full use of historical data and propose predicting the interception point and its interception time in a data-driven form so as to achieve efficient penetration.
2. An HHO-CNN-SVR interception point and interception time prediction model is proposed for the first time. This model can process historical data features more efficiently, obtain hyperparameters for the algorithm more accurately, and achieve accurate prediction.
3. The dataset constructed by the attack and defense model verifies that the HHO-CNN-SVR algorithm has high accuracy and stability.

The structure of this paper is as follows: the second section puts forward the problems to be solved in this paper, the third section explains the method of constructing the dataset in this study, the fourth section is the proposed prediction algorithm model, the fifth section describes the simulation configuration, and the last section is the simulation results and conclusions.

2. Problem Description

With the rapid development of modern air defense technology, the difficulty of preventing missiles from being intercepted is increasing. Missiles must adopt multiple avoidance strategies during flight to deal with interception threats at different stages. In addition, missiles face severe challenges in the field of electronic warfare. Modern air defense systems not only rely on radar tracking; they can also destroy the accuracy of navigation and target recognition of missiles by means of electronic interference, false targets, and antimunition strategies. Some high-precision intercept missiles can achieve accurate hits in a very short time, which requires the attacking missile to constantly adjust the flight trajectory, speed,

and even height during flight, thereby increasing the probability of success in breaking through the air defense network.

It can be seen from Figure 1 that predicting the intercept point in advance helps to formulate an effective penetration strategy. This prediction provides a valuable time window for the missile to take appropriate avoidance or countermeasures before approaching the intercept point to maximize the probability of successful penetration. Therefore, the prediction of the interception point and interception time is the key to efficient penetration.

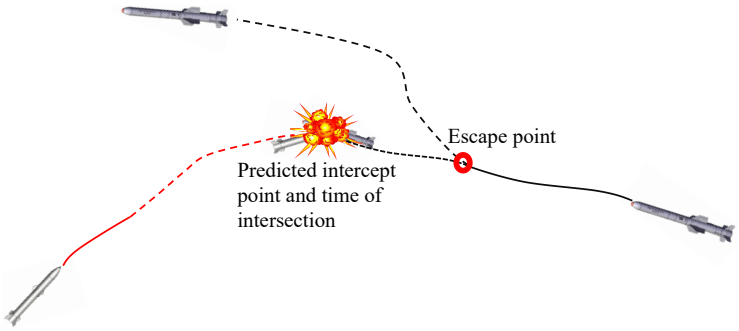


Figure 1. Missile penetration schematic diagram.

3. Simulated Data

Supervised learning and unsupervised learning are two basic paradigms of machine learning. The difference between them is whether the data are labeled. Supervised learning is trained using labeled data. The goal is to learn the relationships between the input and output, which is suitable for classification and regression tasks [30]. Unsupervised learning deals with unlabeled data, mainly by discovering the internal structure of the data to perform tasks such as clustering or dimensionality reduction.

In the modern war environment, the combat cycle is generally long, and the application of guided projectiles is increasingly widespread. Because these weapons are frequently launched in combat, they can provide a large amount of actual combat data for subsequent tactical analysis and decision making. In addition, after the launch of our guided projectiles, the enemy is bound to intercept them. Collecting the information from the guided projectiles and the enemy interceptors successfully intercepted by the enemy as well as analyzing the potential relationship between their position and attitude and the interception time and interception point are conducive to our offensive missiles achieving penetration.

Therefore, this study adopted the supervised learning method to obtain experience from cases of attack failure (intercepted missile), so as to realize online prediction.

3.1. Dataset Construction

In the modern combat environment, the position of the interceptor and the offensive projectile is relatively easy to obtain, and the speed and attitude information of our offensive projectile is also easy to obtain. Due to the limitation of experimental conditions, we were not able to obtain real experimental data. In reality, the acquisition channels of offensive and defensive data are diverse, such as for the data collected in past combat processes, military drill data and interception, penetration, and other experimental process data, so real data available from training must exist. For the first time, this study proposes using machine learning to predict interception points. In order to make our method more convincing, the data were obtained by simulating an attack and defense scene. In order to verify the feasibility of the interception time and interception point prediction method proposed in this paper, the motion relationship of the missile intercepting the maneuvering target is shown in Figure 2.

In the figure, OXY is the inertial coordinate system, where the missile is located at point M, the target is located at point T, r is the relative distance between the missile and the target, x_B is the longitudinal axis direction of the projectile, and v_M and a_M are the velocity and normal acceleration of the missile, respectively. V_T and a_T are the velocity and normal acceleration of the target, respectively. θ_M and θ_T are the velocity inclination angles of the missile and the target, respectively; q is the line-of-sight angle of the projectile; ϑ is the pitch angle, where the above angles are positive in the counterclockwise direction.

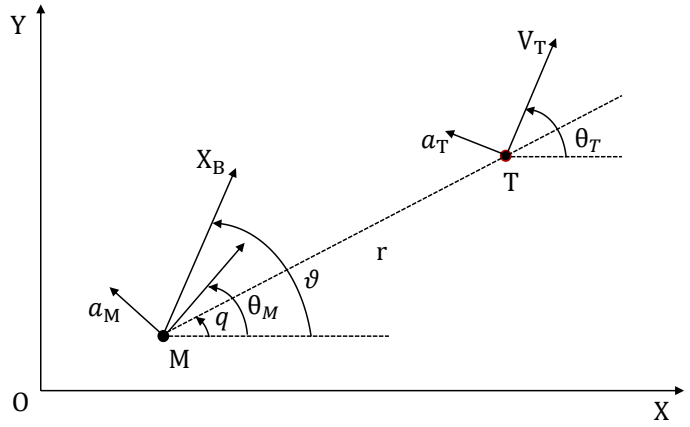


Figure 2. Missle–target relative motion model.

Firstly, based on the knowledge of flight mechanics, the relative motion equation and the dynamic equation of the projectile are obtained.

The kinematic equations for missile–target motion in the pitch plane are as follows:

$$\begin{cases} \dot{r} = V_T \cos(\theta_T - q) - V_M \cos(\theta_M - q) \\ \dot{q} = \frac{V_T \sin(\theta_T - q) - V_M \sin(\theta_M - q)}{r} \\ \dot{\theta}_M = \frac{a_M}{V_M} \\ \dot{\theta}_T = \frac{a_T}{V_T} \end{cases} \tag{1}$$

The dynamics equations for the missile–target system in the pitch plane are as follows:

$$\begin{cases} a_M = n_y g = \frac{Y_0}{m} - g\cos(\theta_M) \\ a = \vartheta - \theta_M \\ \dot{\vartheta} = \omega_z \\ \dot{\omega}_z = \frac{M_0}{J_z} \end{cases} \tag{2}$$

For the detailed process of parameter definition, parameter configuration, and derivation of control input in the formula, refer to the article [31]. Specifically, the sliding mode surface, the reaching law, the virtual control quantity, and the control input are

The sliding surface is

$$s_1 = x_2 + a_0 x_1 + \beta_0 x_1^{q_0/p_0} \tag{3}$$

The reaching law is

$$\dot{s_1} = -\varphi s_1 - \gamma s_1^{\frac{\lambda}{\eta}} \tag{4}$$

The virtual control quantity is

$$\begin{cases} x_{3d} = -b_2^{-1}\left[f_2(x_2) + a_0 x_2 + \beta_0 \frac{q_0}{p_0} x_1^{\frac{q_0-p_0}{p_0}} x_2 + \varphi s_1 + \gamma s_1^{\frac{\lambda}{\eta}} + \hat{d}_2\right] \\ x_{4d} = b_3^{-1}\left[\dot{x}_{3c} + k_3(x_{3c} - x_3) - f_3(x_3) - \hat{d}_3\right] \\ x_{5d} = b_4^{-1}\left[\dot{x}_{4c} + k_4(x_{4c} - x_4) - f_4(x_4) - \hat{d}_4\right] \end{cases} \tag{5}$$

The control input is

$$\delta_{zc} = b_5^{-1}[\dot{x}_{5c} + k_5(x_{5c} - x_5) - f_5(x_5)] \tag{6}$$

As shown in Figure 3, under the action of the control input, the enemy's maneuvering target can be accurately intercepted.

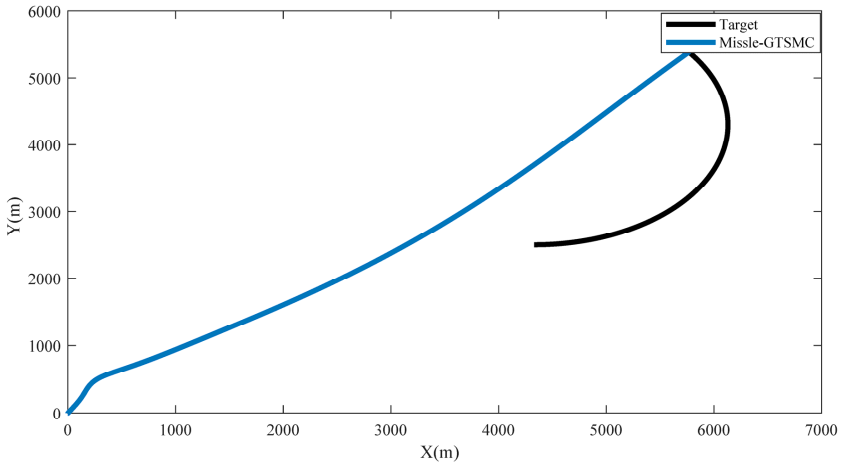


Figure 3. Missile target engagement diagram.

Because the starting position of the interceptor and the target maneuver is uncertain, a random number in the range of $[-400, 400]$ is added to the starting position of the interceptor, according to the literature [31], and a random number in the range of $[-500, 500]$ was added to the starting position of the target maneuver. Moreover, the initial firing Angle and initial speed of the interceptor are constantly adjusted to sample data, in which the sampling method is shown in Table 1.

Table 1. Attack and defense data sampling method.

Parameter	Sampling Value	Sampling Interval
Initial velocity (m/s)	$(600, 900]$	5
Shooting angle (°)	$(40, 70]$	1

Due to the fast flight speed and short flight time of a projectile in the final guidance stage and because less data can be obtained from actual environments, the launch point of the interceptor and the starting point of the maneuvering target are regarded as the first observable value. According to the above sampling method, a total of 1800 sets of offensive and defensive information can be obtained, so the data are saved as $escape_i = (x_m, y_m, x_t, y_t, \theta_m, v_m, X, Y), i = 1, 2, \ldots, 1800$.

3.2. Data Preprocessing

Before training, the dataset should be normalized. Normalization can unify data of different dimensions and ranges into a standard scale, so as to eliminate the differences in

magnitude between the data and avoid the learning process of some feature-dominated models. It helps to accelerate convergence and improve the training efficiency of the model. The formula is as follows:

$$T_N = \frac{T - T_{min}}{T_{max} - T_{min}} \tag{7}$$

T_{min} and T_{max} are the minimum and maximum values of the corresponding data, and T_N is the normalized data.

4. Forecasting Model

The HHO-CNN-SVR model combines Harris Hawk optimization, a convolutional neural network, and support vector regression. High-level features are automatically extracted from the data through the CNN, and then these features are used as the input of the SVR for regression prediction. Different from the traditional SVR model that relies on manual feature selection, CNNs can effectively capture the features in the data that are conducive to the regression problem through the convolution layer and the pooling layer, and they overcome the limitations of traditional SVR in feature selection. However, the performance of SVR is highly dependent on the selection of its hyperparameters. An intelligent optimization algorithm can automatically adjust the hyperparameters of SVR, which can effectively explore the parameter space and avoid the local optimal problem so as to improve the model's performance and generalization ability.

4.1. HHO

HHO (Harris Hawk optimization) is composed of three stages: the exploration stage, the exploitation stage, and the phase discrimination stage. In the basic HHO algorithm, the position of the Harris Hawks is initialized randomly and updated using the following position update mechanism:

1. Exploration phase

During the exploration phase, Harris Hawks adopt the following two strategies to search for prey:

$$X_H\left(t+1\right) = \begin{cases} X_{rand}\left(t\right) - k_1\left|X_{rand}\left(t\right) - 2k_2\, X_H\left(t\right)\right| & q \geq 0.5 \\ X_{rab}\left(t\right) - X_{mean}\left(t\right) - k_3\left(LB(1-k_4) + k_4\, UB\right) & q < 0.5 \end{cases} \tag{8}$$

where $X_H(t)$ is the location of the Harris Hawk at iteration (t), X_{rab} is the location of the optimal individual, LB and UB are the constraints on the search location, $k_i, i = 1, 2, 3, 4, 5$ is a random number between $(0, 1)$, and q is also a random number between $(0, 1)$. However, the size of q directly affects the selection of search methods, and $X_{mean}(t)$ is the average value of individuals in the iterative process.

2. Stage discrimination

From the exploration stage to the development stage, the Harris Hawk optimization mainly relies on the escape energy function of prey to distinguis the escape energy function of the prey:

$$E = 2E_{init}\left(1 - \frac{t}{T_{max}}\right) \tag{9}$$

where T_{max} is the maximum number of iterations, and E_{init} is the initial value of the energy function, which gradually decreases as the number of iterations increases. When $|E| < 1$, the development phase update policy starts, and when $|E| \geq 1$, the exploration phase location update policy starts.

3. Development phase

According to the escape probability r and escape energy E of the prey, the Harris Hawk can formulate four encircling modes:

Method 1: Soft surround

When prey energy is sufficient, the Harris Hawk chooses the soft encircle strategy of exhausting the prey energy first and then rounding them up to attack. The updated strategy is as follows:

$$X_H(t+1) = \Delta X_H(t) - E|X_H(t) - JX_{rab}(t)| \tag{10}$$

$$\Delta X_H(t) = X_{rab}(t) - X_H(t) \tag{11}$$

At this time, the escape probability $r \geq 0.5$, and the escape energy $|E| \geq 0.5$, where the position update strategy of the prey during the escape process is

$$J = 2(1 - k_5) \tag{12}$$

Method 2: Hard Siege

At this time, the prey is low on energy, and the Harries Hawks can take down the prey in one swoop. The siege strategy is as follows:

$$X_H(t+1) = X_{rab}(t) - E|\Delta X_H(t)| \tag{13}$$

In this case, the escape probability $r \geq 0.5$, and the escape energy $|E| < 0.5$.

Mode 3: Progressive dive soft encircle

At this time, the prey is preparing to escape, but the prey has enough energy; the Harris Hawks have to adopt a progressive tracking and capture mode. The progressive siege strategy is

$$Y = X_{rab}(t) - E|X_H(t) - JX_{rab}(t)| \tag{14}$$

When the fitness function does not change, the Harris Hawk adjusts to perform the following strategies:

$$Z = Y + S \times levy(D) \tag{15}$$

where $leavy(x)$ is as follows:

$$levy(x) = 0.01 \times \frac{u \times \partial}{|v|^{\frac{1}{\beta}}} \tag{16}$$

$$\partial = \left(\frac{\Gamma(1+\beta) \times sin(\frac{\pi\beta}{2})}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}}\right) \tag{17}$$

Therefore, the update strategy of Method 3 can be summarized as

$$X_H(t+1) = \begin{cases} Y & F(Y) < F(X_H(t)) \\ Z & F(Z) < F(X_H(t)) \end{cases} \tag{18}$$

In this case, the escape probability r < 0.5, and the escape energy $|E| > 0.5$.

Method 4: Progressive dive hard encircling

When the prey is fleeing and the prey is low on energy, the Harris Hawk can take down the prey at a distance and execute the encircle and attack strategy:

$$X_H(t+1) = \begin{cases} Y & F(Y) > F(X_H(t)) \\ Z & F(Z) > F(X_H(t)) \end{cases} \tag{19}$$

In Equation (19), Y and Z are, respectively,

$$Y = X_{rab}(t) - E|JX_{rab}(t) - X_m(t)| \tag{20}$$

$$Z = Y + S \times levy(D) \tag{21}$$

In this case, the escape probability $r < 0.5$, and the escape energy $|E| < 0.5$.

4.2. CNN

CNN is becoming more and more widely used in data regression prediction, especially in feature extraction, showing its unique importance. CNN can automatically learn hierarchical feature representations from raw data and extract local dependencies through convolution operations, thereby capturing complex patterns and potential laws in the data. In the regression prediction task, CNN significantly reduces the number of training parameters and computational complexity of the model by sharing the design of convolution kernel parameters and local receptive field, and improves the learning ability of important features in the data. This efficient feature extraction mechanism can effectively improve the prediction accuracy and generalization ability of the model when dealing with data with complex spatial or temporal relationships [32,33]. Compared with traditional machine learning methods, CNN gradually constructs a high-level representation of the data through multilevel nonlinear transformation so as to better reveal the potential relationship between the input features and target variables as well as enhance the expressiveness and adaptability in regression tasks.

In the interception point and interception time prediction task, the CNN uses the pooling layer to reduce the feature dimension and discards the irrelevant features to reduce the complexity of the model. Finally, the full connection layer is used for regression analysis. The CNN is mainly composed of an input layer, a convolution layer, a pooling layer, a fully connected layer, and an output layer. Figure 4 depicts the standard CNN model structure.

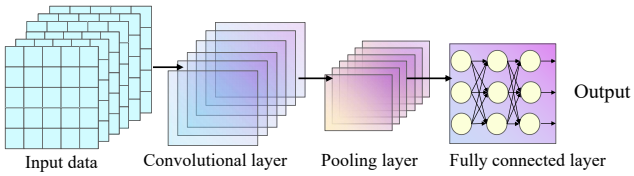


Figure 4. CNN model structure diagram.

4.3. SVR

SVR is a regression analysis method based on support vector machine (SVM) theory. SVR fits the data by constructing a maximum interval regression model, which has good generalization ability and performs well in dealing with nonlinear regression problems.

The goals of SVR are to make the error between the predicted value and the true value of each data point less than a tolerance ϵ by selecting an appropriate function $f(x)$ and to make the regression function as smooth as possible. Support vector regression is described by a linear regression model in the form of

$$f(x) = \langle w, x \rangle + b \tag{22}$$

where w is the weight vector, b is the bias term, and $\langle w, x \rangle$ represents the inner product.

The optimization objective of SVR is to minimize the objective function

$$min_{w,b,\xi_i,\xi_i^*} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{23}$$

and the following constraints need to be met:

$$\begin{cases} y_i - \langle w, x_i \rangle - b \leq \epsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{24}$$

where ξ_i, ξ_i^* are slack variables, and C is a penalty factor, indicating the importance of outliers. A larger C makes the model fit the sample better and the model more complex.

In the actual regression problem, there may be nonlinear inseparable data in the input space. In order to accurately divide the data and find the appropriate regression hyperplane, SVR introduces the kernel function $K(x_i, x_j)$. By mapping the input data from the original low-dimensional space to a higher-dimensional feature space, the data become linearly separable in the mapped feature space. In this way, linear methods can be used to solve nonlinear regression problems in high-dimensional space. In this paper, the radial basis function is selected as the kernel function, and its expression is

$$K(x_i, x_j) = \exp\left(-\frac{|x_i - x_j|^2}{2\sigma^2}\right) \tag{25}$$

Among them, $(|x_i - x_j|)$ is the Euclidean distance between samples x_i and x_j, and σ is the width parameter of the kernel function, which controls the measurement of similarity between samples.

The final regression prediction function of the SVR model can be expressed as

$$y(x) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) K(x, x_i) + d \tag{26}$$

where α_i and α_i^* are variables obtained by the Lagrange multiplier method in the optimization problem, representing the weight of each sample; d is the bias term of the regression model.

5. Simulation Configuration

The simulation experiment was based on the Matlab platform (latest v. R2024b), and the NVIDIA GeForce RTX 4060 graphics card (NVIDIA, Santa Clara, CA, USA) was used for training. The process is shown in Figure 5.

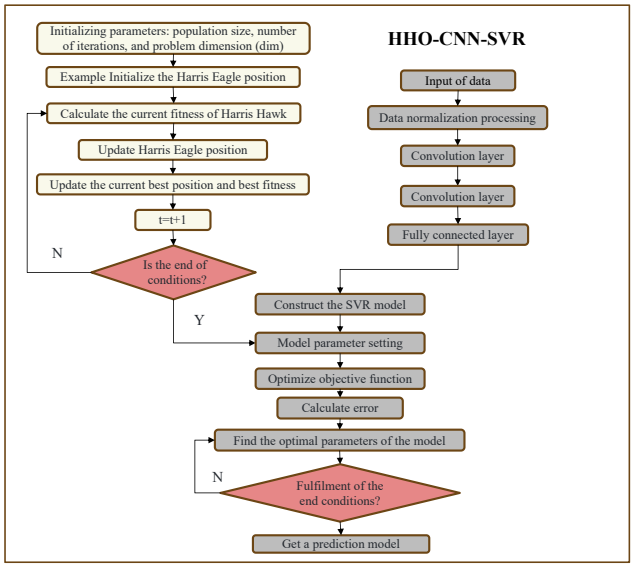


Figure 5. Projectile interception point and interception time prediction flow chart.

In this paper, by combining the superior optimization performance of the HHO algorithm, the powerful feature extraction ability of the CNN, and the advantages of SVR regression prediction, an HHO-CNN-SVR model for the prediction of the projectile interception point and interception time is proposed; we compared the prediction results of the SVR, CNN-SVR, and HHO-CNN-SVR models used in the improvement process. The model parameters and model structure configuration are shown in Table 2:

Table 2. Model parameters and structure configuration.

Model	Parameter	Configuration
BP	Iteration times	500
	Error target	1×10^{-4}
	Learning rate	0.01
	Hidden neurons	10
ELM	Hidden neurons	10
HHO-CNN-SVR	The first convolution layer	Convolution kernel number 16, length of 2
	Second convolution layer	Convolution kernel number 32, length of 2
	Dropout	Dropout rate 0.2
	Fully connected layer	Number of neurons 1
	$-p, -t$	0.01, 2

The SVR and CNN-SVR model configurations were the same as that of the HHO-CNN-SVR model. The number of training times of the CNN network structure was set to 500, and the learning rate was set to 0.01. In the HHO algorithm, the number of populations was set to 30, and the maximum number of iterations was set to 100.

Since the construction of the above data set adopts a two-dimensional offensive and defensive model to obtain, the model input is shown in Table 3:

Table 3. Input layer node configuration.

Input Layer Node						
Abbreviation	x_m	y_m	x_t	y_t	θ_m	v_m
Full name	Range of attack projectile	Attack ejection high	Interceptor range	Interceptor ejection height	Attack projectile velocity inclination	Attack bullet speed

The output term of the interception time prediction was set to the interception time t to be predicted, and the output term of the interception point prediction was set to the interception point position (X, Y) to be predicted.

6. Simulation Result

In this study, the network structure of HHO-CNN-SVR was used to train two models to predict the interception time and interception point, separately. The simulation results were compared with the prediction results of the SVR, HHO-SVR, and CNN-SVR proposed in the process of algorithm improvement. Horizontal, BP, ELM, and other algorithms were compared.

6.1. Interception Time Prediction

After training the prediction model, the data were input into the trained network weight matrix in turn, and the output results of the network were compared with the output values of the original test set to verify the effectiveness of the model. The validity of the model was verified, and evaluation indices including mean value, standard deviation, maximum error, coefficient of determination, root mean square error, and average deviation error [34] were used. The specific expressions are

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y}_i| \tag{27}$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2 \tag{28}$$

$$\mathrm{MAX_E} = max_{i=1}^{n}|y_i - \widehat{y}_i| \tag{29}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{30}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2} \tag{31}$$

$$\mathrm{MBE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i) \tag{32}$$

The interception time prediction results are shown in Table 4.

Table 4. Prediction results of interception time.

Algorithm	MAE	MSE	MAX_E	R^2	RMSE	MBE	TIME (s)
BP	0.037	0.0024	0.24	0.998	0.049	−0.0102	0.00323
ELM	0.044	0.0779	0.81	0.964	0.279	0.0094	0.00048
SVR	0.057	0.0082	0.54	0.996	0.090	0.0123	0.00311
HHO-SVR	0.036	0.0023	0.22	0.998	0.048	−0.0001	0.00169
CNN-SVR	0.028	0.0020	0.21	0.999	0.044	−0.0068	0.00143
HHO-CNN-SVR	0.013	0.0009	0.21	0.999	0.030	0.0013	0.00087

It can be seen from Table 4 that the HHO algorithm can automatically adjust the hyperparameters of SVR and can efficiently explore the parameter space. Compared with the SVR prediction results, the average absolute error was reduced by 36%. The CNN-SVR model automatically extracts high-level features from the data through CNN and then uses these features as the input of SVR for regression prediction. Compared with the SVR prediction results, the average absolute error was reduced by 50%. These two improvement measures effectively improved the model's performance and generalization ability.

The HHO-CNN-SVR model combines HHO, CNN, and SVR to give full play to the powerful optimization ability of CNN feature extraction and HHO. The model predicts the minimum MAE and MSE of the interception time, indicating that the algorithm has good accuracy and stability. The average absolute error of the interception time prediction result was 0.0139 s. Since the penetration process starts a period of time before the interception point, the time between the start time and the interception time is much longer than 0.0139 s, so the error is within the controllable range.

In the horizontal comparison results, although ELM had the fastest prediction speed, its prediction accuracy was far less than that of the HHO-CNN-SVR algorithm proposed in this paper. The coefficient of determination of the HHO-CNN-SVR algorithm was closest to one, and its fitting effect is the best. In order to clearly present the need for algorithm improvement in the sample point prediction error map and the box plot, the ELM and BP algorithm prediction results are not drawn in the sample point prediction error or the box plot.

Figure 6 shows the prediction error results of 50 randomly selected sample points for the interception time. It can be seen from the figure that the prediction results of the SVR, HHO-SVR, and CNN-SVR algorithms were unstable, and the error fluctuated greatly, which would directly affect the decision making regarding offensive projectile penetration. The prediction error of the HHO-CNN-SVR model proposed in this paper was always

in the range of 0.1 s, the error fluctuation was not large, and the prediction result was relatively stable.

According to the box plot of the interception time prediction error in Figure 7, it can be seen that the medians of the SVR, HHO-SVR, and CNN-SVR prediction results are higher, and the number of outliers is lower. HHO-CNN-SVR performed best among all methods, having the lowest median and the least number of outliers as well as achieving more stable prediction results.

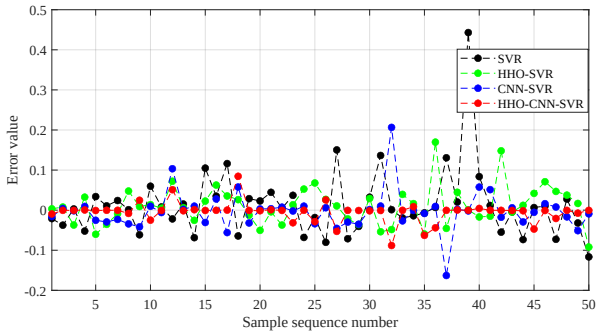


Figure 6. Interception time sample point prediction error.

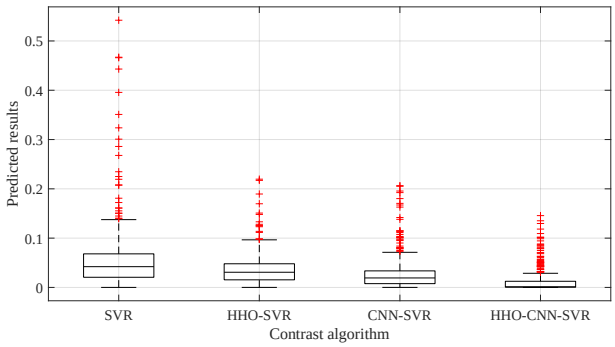


Figure 7. Interception time prediction error box plot.

6.2. Interception Point Prediction

Aside from the interception time prediction, if we can accurately predict where our attack projectile is intercepted and change the control input of the attack projectile before reaching the interception point, we can effectively achieve penetration. At the same time, accurately predicting the location of the interception point can improve the probability of our interceptor intercepting the enemy's offensive projectile and more efficiently destroy the threat posed by the offensive projectile.

It can be seen from Tables 5 and 6 that the HHO-CNN-SVR model had the best prediction effect on the interception point. The prediction error in the direction of the range was only 26% of the prediction error of the SVR model, and the error in the direction of the firing height was only 13% of the prediction error of the SVR model. Its prediction accuracy and stability were greatly improved compared with the HHO-SVR model and the CNN-SVR model. The MAE in the range direction was 2.3 m, and the MAE in the height direction was 2.01 m. The prediction error was very small, which can meet engineering requirements. The location of the interception point could be accurately predicted, and then penetration could be realized.

In the transverse comparison results, although the prediction speed of ELM was still the fastest, the maximum errors in the prediction results in the range and elevation directions were 173 m and 269 m, respectively, and the MSEs were 4409 m and 7841 m, respectively. The

prediction results were relatively unstable and prone to abnormal values. The BP prediction results were far worse than the HHO-CNN-SVR algorithm proposed in this paper.

Table 5. Prediction results of intercept point in range direction.

Algorithm	MAE	MSE	MAX_E	R^2	RMSE	MBE	TIME (s)
BP	7	102	50	0.998	10	−1.13	0.00323
ELM	10	4409	173	0.964	66	−1.70	0.00048
SVR	8	143	58	0.994	11	−0.68	0.00347
HHO-SVR	6	68	33	0.997	8	0.69	0.00343
CNN-SVR	4	61	53	0.998	7	−1.356	0.00232
HHO-CNN-SVR	2	16	19	0.999	4	1.04	0.00210

Table 6. Prediction results of intercept point in shooting height direction.

Algorithm	MAE	MSE	MAX_E	R^2	RMSE	MBE	TIME (s)
BP	13	320	100	0.998	17	−4.4	0.00323
ELM	14	7841	267	0.964	88	2.8	0.00048
SVR	15	419	122	0.998	20	−2.1	0.00529
HHO-SVR	13	368	137	0.998	19	−0.4	0.00385
CNN-SVR	8	188	99	0.999	13	0.5	0.00243
HHO-CNN-SVR	2	19	33	0.999	4	1.1	0.00051

Figure 8 shows the prediction error results of 50 randomly selected intercept point samples. It can be seen from the prediction results in Figure 8a,c that for randomly selected intercept point samples, in the prediction results of the HHO-CNN-SVR model, only one sample had an error of more than 10 m in the range direction, but it was close to 10 m. In the direction of the shooting height, only one sample point had a prediction error of more than 20 m, and the remaining sample points were basically stable near 0, having the smallest prediction error and relatively stable. It can be seen from the box plot of the prediction error in Figure 8b,d that HHO-CNN-SVR performed best among all methods, having the lowest median and the least number of outliers, which could achieve more stable prediction results.

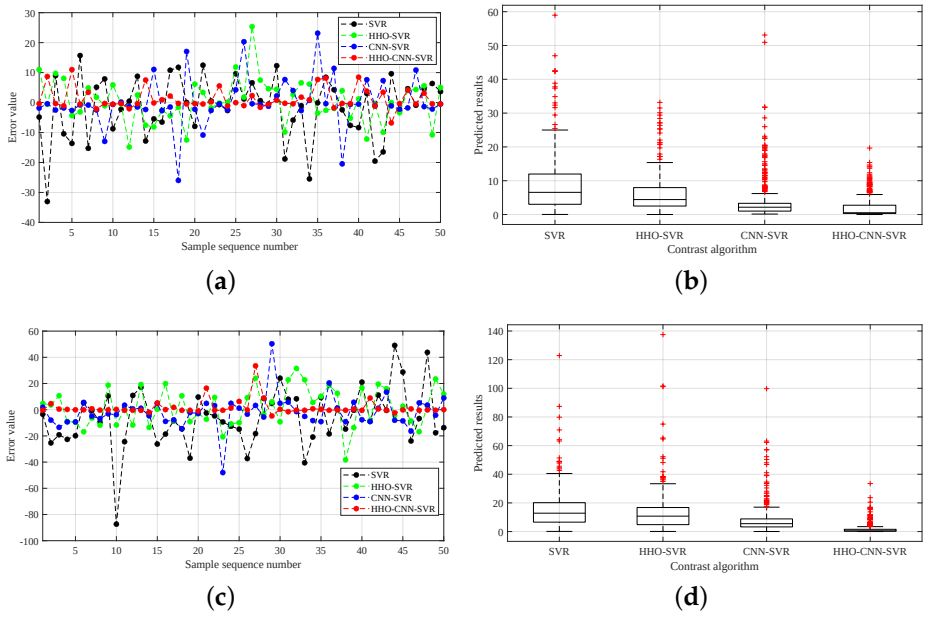


Figure 8. Prediction results of interception point. (**a**) Prediction error of sample points in x-axis direction; (**b**) x-axis direction prediction error box plot; (**c**) prediction error of sample points in y-axis direction; (**d**) y-axis direction prediction error box plot.

6.3. Generalization Performance Test

In the early stage of an operation, due to the lack of attack and defense data in the current combat environment, it is difficult to achieve data-driven interception point and interception time prediction. However, we had data from historical battlefields and military exercises. Therefore, we regarded all the data generated in the second section as our existing data and used these data to train the model. In order to verify that the model has a certain generalization ability, can deal with different types of missiles with different control strategies, and can adapt to different combat environments, the control strategy mentioned in the second section was replaced to generate attack and defense data, and the control strategy was replaced with the following:

1. Based on the bias proportional guidance control method, the virtual control instruction was designed as follows:

$$x_{3d} \cdot g = NV_M\dot{q} + K_a(N-1)\frac{V_M^2}{r}\left[\frac{Nq-\theta_M}{N-1} - q_d\right] \tag{33}$$

2. Based on the PIDSMC control strategy, the virtual instruction was designed as follows:

$$x_{3b} = -\frac{1}{k_d}b_2^{-1}\left[k_d\left(f_2 + \hat{d}_2\right) + k_p x_2 + k_i x_1 - \dot{s}_1\right] \tag{34}$$

where: $k_p = 1.13, k_i = 1.03, k_d = 1.35, N = 3, K_a = 10$.

Under these two control strategies, a random number in the range of [−400, 400] was added to the starting position of the interceptor, a random number in the range of [−500, 500] was added to the starting position of the target maneuver. Moreover, the initial firing Angle and initial velocity of the interceptor are constantly adjusted to sample data, as shown in Table 7.

Table 7. Attack and defense data sampling method.

Parameter	Sampling Value	Sampling Interval
Initial velocity (m/s)	(600, 900]	30
Shooting angle (°)	(40, 70]	5

The missiles in the above two control strategies were simulated according to the above sampling methods. A total of 120 sets of data were generated, $escape_{NEW} = (x_m, y_m, x_t, y_t, \theta_m, v_m, X, Y), NEW = 1, 2, \ldots, 120$. In order to adapt to the unknown complex environment of the battlefield, the above data were disrupted, and the data were input into the trained model for prediction.

The interception time prediction results are shown in Table 8.

Table 8. Prediction results of interception time.

Algorithm	MAE	MSE	MAX_E	R^2	RMSE	MBE	TIME (s)
BP	0.284	0.179	1.2	0.443	0.42	0.282	0.051393
ELM	0.023	0.283	1.8	0.429	0.53	0.290	0.000308
SVR	0.297	0.167	1.2	0.826	0.40	0.245	0.000316
HHO-SVR	0.287	0.179	1.3	0.921	0.42	0.278	0.000575
CNN-SVR	0.037	0.003	0.3	0.998	0.05	−0.002	0.000958
HHO-CNN-SVR	0.022	0.001	0.1	0.999	0.03	−0.005	0.000504

The intercept point prediction results are shown in Tables 9 and 10.

The dataset constructed above was used as historical data, and the data generated by the replacement control strategy in this section were regarded as the relevant data of a new penetration object in a new combat environment, so it was regarded as a new prediction set.

Table 9. Prediction results of intercept point in range direction.

Algorithm	MAE	MSE	MAX_E	R^2	RMSE	MBE	TIME (s)
BP	310	233,891	1259	0.44312	483	298	0.051393
ELM	18	245,691	1257	0.42967	495	297	0.000308
SVR	311	234,647	1263	−0.33737	484	298	0.000301
HHO-SVR	309	234,224	1253	−0.33496	483	298	0.000549
CNN-SVR	30	7548	844	0.83293	86	−3	0.001149
HHO-CNN-SVR	15	370	83	0.99206	19	−12	0.001104

Table 10. Prediction results of intercept point shooting in height direction.

Algorithm	MAE	MSE	MAX_E	R^2	RMSE	MBE	TIME (s)
BP	214	88059	536	0.44312	296	−179	0.05139
ELM	12	86596	530	0.42967	294	−170	0.00030
SVR	218	88283	680	0.70649	297	−180	0.00023
HHO-SVR	210	85423	581	0.716	292	−176	0.00057
CNN-SVR	13	326	62	0.99863	18	5	0.00059
HHO-CNN-SVR	9	141	45	0.99944	11	4	0.00386

From Tables 8–10, it can be seen that BP, ELM, SVR and HHO-SVR performed poorly on the new prediction set. In the interception point prediction results, the range and elevation direction errors reached 1200 m and 500 m, respectively. After adding CNN, the generalization ability of the prediction was greatly improved, mainly because the CNN could effectively extract the local features in the data and capture more complex patterns through multiple convolutional layers, making the model more robust in the face of unknown data and better performing feature learning and generalization. The MAEs of the HHO-CNN-SVR algorithm for the prediction of the interception point in the range and elevation directions were 15 m and 9 m, respectively, and the prediction results meet engineering requirements. Therefore, the HHO-CNN-SVR model proposed in this paper has both high prediction accuracy and strong generalization ability.

6.4. Robustness Test

In a real combat environment, the combat scene is complex, and there is a certain error in the observed data. By adding noise to the new data calculated in the previous section, the robustness and generalization ability of the model could be verified at the same time. The noise disturbance tests the adaptability of a model in the face of uncertainty in the input data. The better the performance, the stronger the robustness. In the generalization performance test in the previous section, the HHO-CNN-SVR algorithm showed the best generalization ability, while the prediction results of the other algorithms were extremely poor. Therefore, this section directly tests the prediction ability of the HHO-CNN-SVR algorithm in dealing with new data with noise. Due to the large difference in data sizes in the input data, the noise addition method was $(1 + \mathcal{N}(0, \sigma^2) \times$ original data.

It can be seen from Table 11 that the HHO-CNN-SVR algorithm showed strong adaptability in the face of noise disturbance of input data and could maintain high prediction accuracy. The evaluation indices (such as MAE, MSE, MAX_E, R^2, RMSE, etc.) in Table 11 show that the prediction results of the algorithm were ideal, indicating that the HHO-CNN-SVR algorithm has strong robustness as well as excellent generalization ability and is suitable for application in complex and uncertain combat environments.

Table 11. Prediction results.

	MAE	MSE	MAX_E	R^2	RMSE	MBE	TIME (s)
t	0.03	0.002	0.15	0.99	0.045	0.014	0.0004
X	15	322	41	0.99	17.954	−13.346	0.0011
Y	12	279	78	0.99	16.7138	10.477	0.0003

7. Conclusions

This paper mainly focused on the prediction of the interception point and interception time of an interceptor in the process of missile penetration. Due to the large number of input features of sample points, direct modeling may affect the overall prediction performance. Therefore, CNN was used to automatically extract high-level features from the data, and then these features were used as the input of SVR for regression prediction. Considering that the performance of SVR is highly dependent on the selection of its hyperparameters, the HHO algorithm was used to optimize its hyperparameters, and the HHO-CNN-SVR model was proposed. Compared with SVR, HHO-SVR, and CNN-SVR, the HHO-CNN-SVR model has outstanding performance in prediction accuracy and stability, especially having the smallest error for interception time, and the error fluctuation was small. The MAE of the prediction result was only 0.0139 s; in the interception point prediction, the error in the range and elevation directions was significantly lower than that of the comparison models. The MAE in the range direction was 2.3 m, and the MAE in the elevation direction was 2.01 m, which meet engineering requirements. Overall, the HHO-CNN-SVR model has strong prediction accuracy and stability. In addition, different control strategies were used to construct a new prediction set, and noise was added to the prediction set. The HHO-CNN-SVR algorithm maintained good prediction results. The results showed that the HHO-CNN-SVR model proposed in this study has strong generalization ability and high robustness, which can provide reliable support for penetration decision making and defense system optimization.

Regarding future prospects, this research is based on a data-driven method to predict the interception point and interception time, and the model proposed in this paper has strong generalization ability and high robustness. However, with the development of hypersonic and high-maneuverability weapons, although there is an opportunity to predict the position of the interception point for penetration, due to the high maneuverability of the interceptor, it is still possible to track our offensive projectiles. Achieving multiple penetrations is an urgent problem to be solved. At the same time, due to sensor limitations, accurately observing enemy interceptor information is also very challenging.

Author Contributions: Conceptualization, Z.G.; project administration, W.Y. All authors have read and agreed to the published version of this manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (grant number: 62203191), the National Defence Key Laboratory of Science and Technology (grant number: 2022JCJQLB06105), and the Basic Science (Natural Science) Research Projects in Higher Education Institutions (grant number: 22KJB590001).

Data Availability Statement: Data are contained within this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. You, H.; Chang, X.; Zhao, J. Three-dimensional line-of-sight-angle-constrained leader-following cooperative interception guidance law with prespecified impact time. *Chin. J. Aeronaut.* **2025**, *38*, 103151. [CrossRef]
2. Wang, N.; Wang, X.; Bin, E.; Li, Y. Three-dimensional adaptive dynamic surface guidance law for missile with terminal angle and field-of-view constraints. *ISA Trans.* **2024**, *154*, 113–131. [CrossRef]
3. Surve, P.; Maity, A.; Kumar, S.R. Polynomial shaping based three-dimensional impact angle and field-of-view constrained guidance. *Aerosp. Sci. Technol.* **2024**, *147*, 109018. [CrossRef]
4. Fei, D.; Xiaoyu, Z.; Kunpeng, H.; Panlong, T. A new three-dimensional adaptive sliding mode guidance law for maneuvering target with actuator fault and terminal angle constraints. *Aerosp. Sci. Technol.* **2022**, *131*, 107974.
5. Ren, J.; Wu, X.; Liu, Y.; Ni, F.; Bo, Y.; Jiang, C. Long-term trajectory prediction of hypersonic glide vehicle based on physics-informed transformer. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *59*, 9551–9561. [CrossRef]
6. Wei, C.Y. Dynamic interception point guidance algorithm based on particle swarm optimization. *Meas. Control* **2022**, *55*, 983–995.
7. Yu, J.; Shi, Z.; Dong, X.; Li, Q.; Lv, J.; Ren, Z. Impact time consensus cooperative guidance against the maneuvering target: Theory and experiment. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *59*, 4590–4603. [CrossRef]
8. Chen, Z.; Chen, W.; Liu, X.; Cheng, J. Three-dimensional fixed-time robust cooperative guidance law for simultaneous attack with impact angle constraint. *Aerosp. Sci. Technol.* **2021**, *110*, 106523. [CrossRef]
9. Dong, W.; Wang, C.; Wang, J.; Xin, M. Three-dimensional nonsingular cooperative guidance law with different field-of-view constraints. *J. Guid. Control. Dyn.* **2021**, *44*, 2001–2015. [CrossRef]
10. Li, Y.; Zeng, H.; Zhang, M.; Wu, B.; Qin, X. Global de-trending significantly improves the accuracy of xgboost-based county-level maize and soybean yield prediction in the midwestern united states. *Giscience Remote Sens.* **2024**, *61*, 2349341. [CrossRef]
11. Qiu, X.; Zhang, R.; Yang, J. Prediction of bof endpoint carbon content and temperature via cssa-bp neural network model. *J. Iron Steel Res. Int.* 2024, *prepublish*. [CrossRef]
12. Gao, Z.; Yi, W. Optimizing projectile aerodynamic parameter identification of kernel extreme learning machine based on improved dung beetle optimizer algorithm. *Measurement* **2025**, *239*, 115473. [CrossRef]
13. Wang, Y.; Sun, S.; Chen, X.; Zeng, X.; Kong, Y.; Chen, J.; Guo, Y.; Wang, T. Short-term load forecasting of industrial customers based on svmd and xgboost. *Int. J. Electr. Power Energy Syst.* **2021**, *129*, 106830. [CrossRef]
14. Li, J.; Ye, M.; Wang, Y.; Wang, Q.; Wei, M. A hybrid framework for predicting the remaining useful life of battery using gaussian process regression. *J. Energy Storage* **2023**, *66*, 107513. [CrossRef]
15. Zong, Y.; Nian, Y.; Zhang, C.; Tang, X.; Wang, L.; Zhang, L. Hybrid grid search and bayesian optimization-based random forest regression for predicting material compression pressure in manufacturing processes. *Eng. Appl. Artif. Intell.* **2025**, *141*, 109580. [CrossRef]
16. Li, S.; Cai, T.T.; Li, H. Transfer Learning for High-Dimensional Linear Regression: Prediction, Estimation and Minimax Optimality. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2021**, *84*, 149–173. [CrossRef]
17. Lin, G.; Lin, A.; Gu, D. Using support vector regression and k-nearest neighbors for short-term traffic flow prediction based on maximal information coefficient. *Inf. Sci.* **2022**, *608*, 517–531.. [CrossRef]
18. Heilemann, J.; Klassert, C.; Samaniego, L.; Thober, S.; Marx, A.; Boeing, F.; Klauer, B.; Gawel, E. Projecting impacts of extreme weather events on crop yields using lasso regression. *Weather Clim. Extrem.* **2024**, *46*, 100738.. [CrossRef]
19. Lin, Z. Enhanced gru-based regression analysis via a diverse strategies whale optimization algorithm. *Sci. Rep.* **2024**, *14*, 25629. [CrossRef]
20. Shoaei, M.; Kowsary, F. Multi-objective optimization of geometrical parameters of a pipe with internal baffle using artificial neural networks and genetic algorithm. *Appl. Therm. Eng.* **2025**, *262*, 125212. [CrossRef]
21. Yu, P.; Zhou, C.; Yu, Y.; Chang, Z.; Li, X.; Huang, K.; Yu, J.; Yan, K.; Jiang, X.; Su, Y. Improved pso-tcn model for soh estimation based on accelerated aging test for large capacity energy storage batteries. *J. Energy Storage* **2025**, *108*, 115031. [CrossRef]
22. Li, X.; Zhou, S.; Wang, F.A cnn-bigru sea level height prediction model combined with bayesian optimization algorithm. *Ocean Eng.* **2025**, *315*, 119849. [CrossRef]
23. Khajavi, H.; Rastgoo, A. Improving the prediction of heating energy consumed at residential buildings using a combination of support vector regression and meta-heuristic algorithms. *Energy* **2023**, *272*, 127069. [CrossRef]
24. Hua, L.; Zhang, C.; Peng, T.; Ji, C.; Nazir, M.S. Integrated framework of extreme learning machine (elm) based on improved atom search optimization for short-term wind speed prediction. *Energy Convers. Manag.* **2022**, *252*, 115102. [CrossRef]
25. Shariati, M.; Mafipour, M.; Ghahremani, B.; Azarhomayun, F.; Ahmadi, M.; Nguyen-Thoi, T.; Shariati, A. A novel hybrid extreme learning machine–grey wolf optimizer (elm-gwo) model to predict compressive strength of concrete with partial replacements for cement. *Eng. Comput.* **2022**, *38*, 757–779. [CrossRef]
26. Kardani, N.; Bardhan, A.; Samui, P.; Nazem, M.; Zhou, A.; Armaghani, D.J. A novel technique based on the improved firefly algorithm coupled with extreme learning machine (elm-iff) for predicting the thermal conductivity of soil. *Eng. Comput.* **2021**, *38*, 3321–3340. [CrossRef]

27. Ji, C.; Zhang, C.; Hua, L.; Ma, H.; Nazir, M.S.; Peng, T. A multi-scale evolutionary deep learning model based on ceemdan, improved whale optimization algorithm, regularized extreme learning machine and lstm for aqi prediction. *Environ. Res.* **2022**, *215*, 114228. [CrossRef]
28. Saleh, A.L.; Amyal, F.A.; Számel, L. An enhanced current chopping control strategy for srm drives using harris hawks optimization algorithm. *ISA Trans.* **2024**, *150*, 338–358. [CrossRef]
29. Sakai, K.; Odonchimed, S.; Takano, M.; Takahashi, H. Automated design of digital filters using convolutional neural networks for extracting ringdown gravitational waves. *Mach. Learn. Sci. Technol.* **2024**, *5*, 045043. [CrossRef]
30. Gui, J.; Chen, T.; Zhang, J.; Cao, Q.; Sun, Z.; Luo, H.; Tao, D. A survey on self-supervised learning: Algorithms, applications, and future trends. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 9052–9071. [CrossRef]
31. Li, Z.; Dong, Q.; Zhang, X.; Gao, Y. Impact angle-constrained integrated guidance and control for supersonic skid-to-turn missiles using backstepping with global fast terminal sliding mode control. *Aerosp. Sci. Technol.* **2022**, *122*, 107386. [CrossRef]
32. Wang, S.; Zhang, Y. An attention-based cnn model integrating observational and simulation data for high-resolution spatial estimation of urban air quality. *Atmos. Environ.* **2025**, *340*, 120921. [CrossRef]
33. Tang, Z.; Ji, T.; Kang, J.; Huang, Y.; Tang, W. Learning global and local features of power load series through transformer and 2d-cnn: An image-based multi-step forecasting approach incorporating phase space reconstruction. *Appl. Energy* **2025**, *378*, 124786. [CrossRef]
34. Asgarkhani, N.; Kazemi, F.; Jankowski, R. Machine learning-based prediction of residual drift and seismic risk assessment of steel moment-resisting frames considering soil-structure interaction. *Comput. Struct.* **2023**, *289*, 107181. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.