



MP-KAN: An effective magnetic positioning algorithm based on Kolmogorov-Arnold network

Zibo Gao, Ming Kong^{*}

College of Metrology and Measurement Engineering, China Jiliang University, Hangzhou 310018, China

ARTICLE INFO

Keywords:

Magnetic positioning
Machine learning
Deep learning
Regression forecasting
Neural network
Multilayer perceptron
Kolmogorov-Arnold network

ABSTRACT

Magnetic Positioning (MP) technology represents a novel approach to locating spatial particles, notably medical capsules, wherein the inherently weak and susceptible-to-interference magnetic signals pose stringent demands on spatial positioning algorithms. Traditional methods are usually limited to polynomial fitting, which limits the generalization of the algorithm and the positioning accuracy of the near field part. In this paper, we introduce a magnetic positioning algorithm grounded in the Kolmogorov-Arnold network (MP-KAN), innovatively introduces the neural network method into the magnetic positioning system, providing a novel research idea for the positioning algorithm. Distinguishing from the learnable weight parameters inherent in the traditional model, the KAN network introduces a learnable activation function formulated through spline functions. This innovation enhances model accuracy by leveraging multiple spline curves and executing summation operations at nodes to facilitate regression predictions. Furthermore, the residual of the predicted position and the L1-parameter in the KAN layer and its entropy regularization are used as the prediction loss, and thresholds are strategically set at the network nodes to enhance the generalization ability of the model and obtain the optimal configuration. The proposed method achieves the goal of improving the positioning accuracy of the system while ensuring that the algorithm has a nearly constant positioning accuracy regardless of the distance between the target and the measurement system. The results of an experimental demonstrate that the maximum positioning error within the data set stands at 0.24 mm, the maximum relative error is 5.72 %, the minimum relative error is 0.25 %.

1. Introduction

As a positioning technique offering diverse advantages, including non-contact detection, high measurement precision, and exceptional portability [1], magnetic positioning has garnered considerable attention across various disciplines, such as healthcare, geoscience, and exploration engineering [2–5]. In recent years, research in magnetic positioning has primarily bifurcated into passive and active positioning methodologies, with passive positioning schemes attracting broader interest due to their stable magnetic field intensity and compact size [6]. However, relying solely on idealized models, like the magnetic dipole model, in passive positioning does not guarantee positioning accuracy, particularly in the near-field region of permanent magnets. Consequently, optimization algorithms are frequently employed to enhance positioning precision [7]. C. Hu and colleagues mitigated the limitations of the magnetic dipole model through the application of linear and nonlinear algorithms, reducing the positioning error from 30 mm to 3.8 mm [8–10]. B. Ye and team optimized the artificial bee algorithm,

achieving improved accuracy compared to the particle swarm optimization (PSO) algorithm, and decreasing the positioning error to 4 mm [11]. Q. Fu utilized empirically derived multiple fitting approximations to decrease the positioning error of the magnetic positioning system in the plane to 1.94 mm [12]. W. Song and co-authors employed a differential optimization algorithm for dynamic prediction of position and attitude, achieving a mean square error of 0.37 mm for the spatial Z-axis trajectory prediction [13].

Despite these advancements, these methods often necessitate a substantial number of magnetic sensors and stringent experimental conditions for data acquisition, resulting in zones of inaccurate positioning and posing challenges to precise system localization [14,15]. In response, numerous researchers have recently turned to integrating machine learning (ML) or deep learning (DL) to construct regression models using collected spatial magnetic field data. In recent years, these methods have frequently been extended and applied to tackle traditional engineering problems, often yielding superior performance [16–18]. This is no exception in the realm of localization algorithms, as

* Corresponding author.

E-mail address: mkong@cjlu.edu.cn (M. Kong).

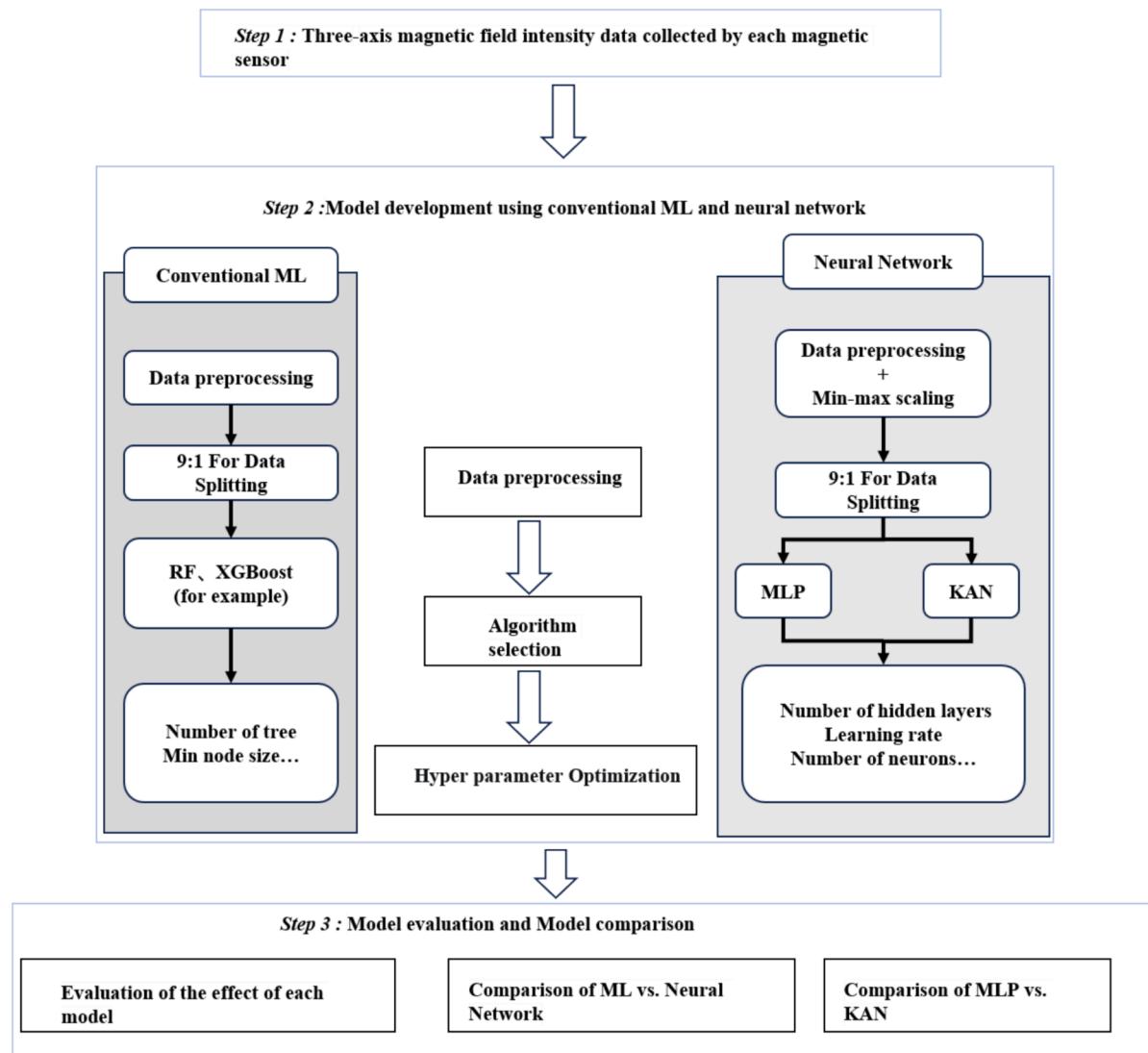


Fig. 1. Workflow of data collection (Step 1), model development (Step 2), and model evaluation and comparison (Step 3).

exemplified by A.-i. Sasaki and E. Ohta demonstrated that within the experimental spatial range, machine learning predictions outperformed traditional localization methods at 73 % of the tested positions [19]. W. He and colleagues conducted a comparative analysis using three ML models—SVM, Random Forest—all of which maintained a high recognition rate for classifying different shapes of permanent magnets [20]. B. Lv and team introduced a method that divides the region and combines the Levenberg-Marquardt algorithm (PSO-LM) with a prior knowledge-based backpropagation neural network (PKBPNN) to enhance positioning accuracy and detection range [21]. Q. Fu presented a regression prediction method based on a backpropagation neural network (BPNN), achieving a positioning rate of 95.2 %, compared to 46.7 % with a nonlinear positioning algorithm, thereby significantly improving the system's positioning rate [22]. Collecting data and training neural networks have proven to significantly bolster positioning accuracy, suggesting that this approach will increasingly become the predominant research trend for addressing magnetic positioning challenges in the future.

Inspired by the advantages and disadvantages of neural networks and combined with the distribution characteristics of magnetic field intensity of permanent magnets in space, this paper proposes a magnetic positioning model based on the Kolmogorov-Arnold network (KAN) [23], which can fully utilize the spatial magnetic field characteristics of

the measured magnet and accurately reconstruct the spatial position of the magnet. Compared with the traditional multi-layer perceptron model (MLP), KAN sets learnable activation functions on the model weights and improves prediction accuracy by transforming each weight parameter into a spline function. In this study, a magnetic positioning test platform was selected for research, and a 10 mm × 10 mm neodymium-iron-boron permanent magnet magnetized along the positive direction of the Z-axis was tested three times at 153 points within a test range of 80 mm × 160 mm. The experimental results show that KAN outperforms MLP and several representative ML algorithms in terms of mean square error and coefficient of determination. In terms of network structure, KAN with a smaller scale can achieve better accuracy compared to MLP, which requires more parameters. The contributions of this paper can be summarized in three aspects:

Initially, we designed a magnetic positioning experimental platform to mimic real-world application scenarios of magnetic positioning, which then served as the foundation for planning the experimental content of this paper.

Subsequently, we constructed a magnetic positioning model named MP-KAN, leveraging the KAN algorithm. A tailored training strategy was implemented to refine the model, optimize its hyperparameters, enhance computational efficiency, and boost its prediction accuracy.

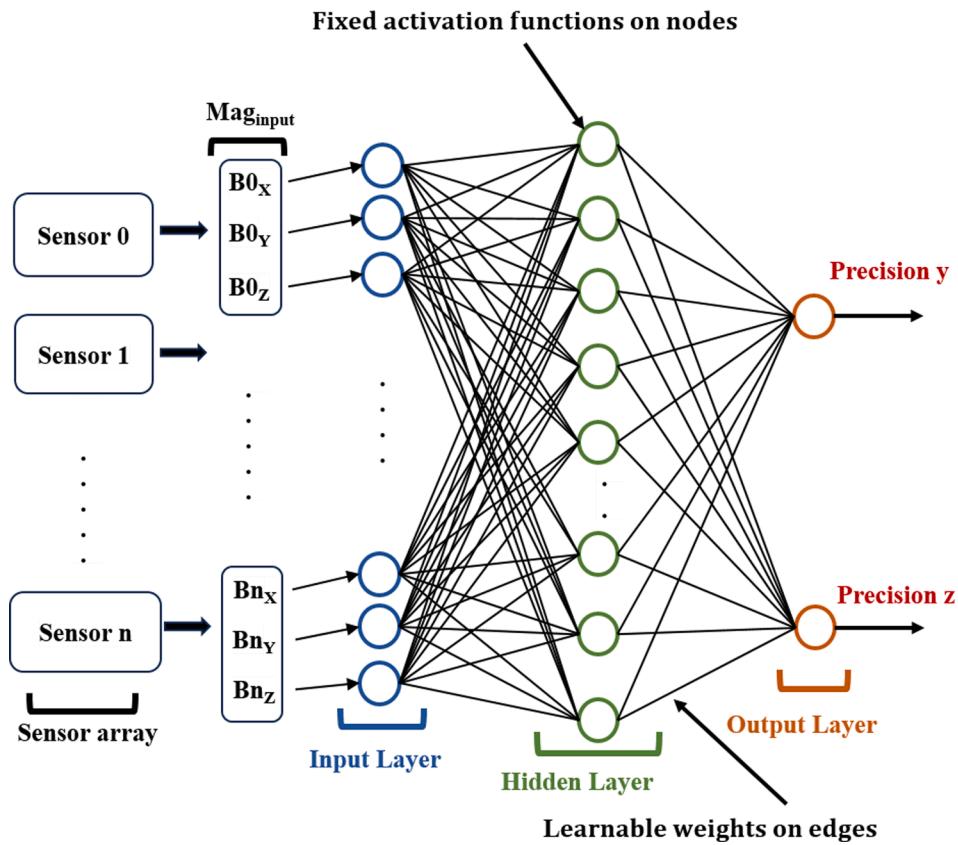


Fig. 2. Three-Layer Multi-Layer Perceptron Architecture.

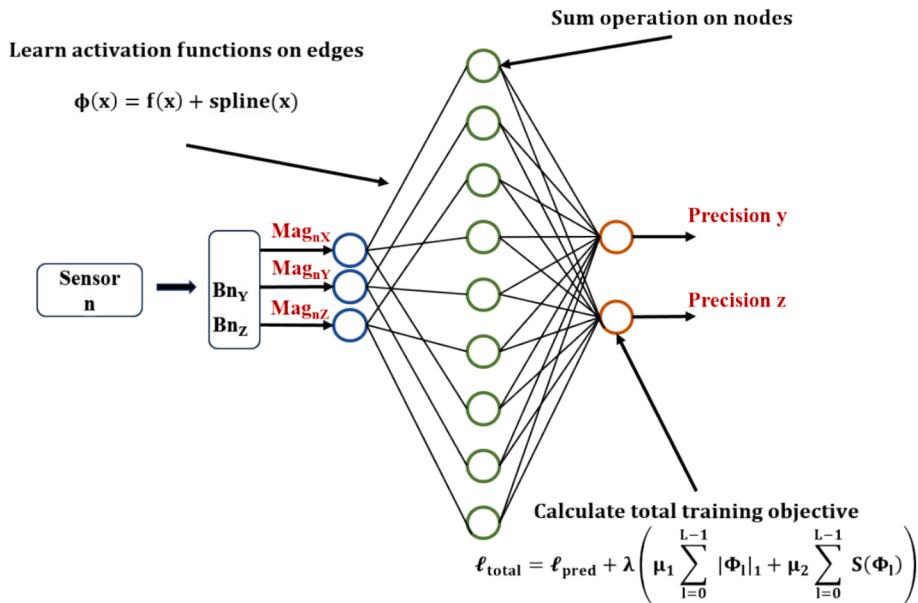


Fig. 3. Three-Layer KAN Architecture for Magnetic Localization Systems.

Ultimately, to assess the performance of MP-KAN, we utilized the same dataset to train a Multi-Layer Perceptron (MLP) and several other representative Machine Learning (ML) models. The results demonstrated that MP-KAN achieves superior positioning accuracy compared to these models, thereby confirming the viability of applying neural network methodologies within magnetic positioning systems.

2. Model structure and training settings

In this section, we propose the MP-KAN algorithm, elucidate its underlying model principles, analyze its distinctions from the MP-MLP model, and select appropriate training settings. The specific workflow of the MP-KAN algorithm is illustrated in Fig. 1. Section 3 delves into the construction of the experimental platform and the experimental design employed in this article. Section 4 presents the prediction results

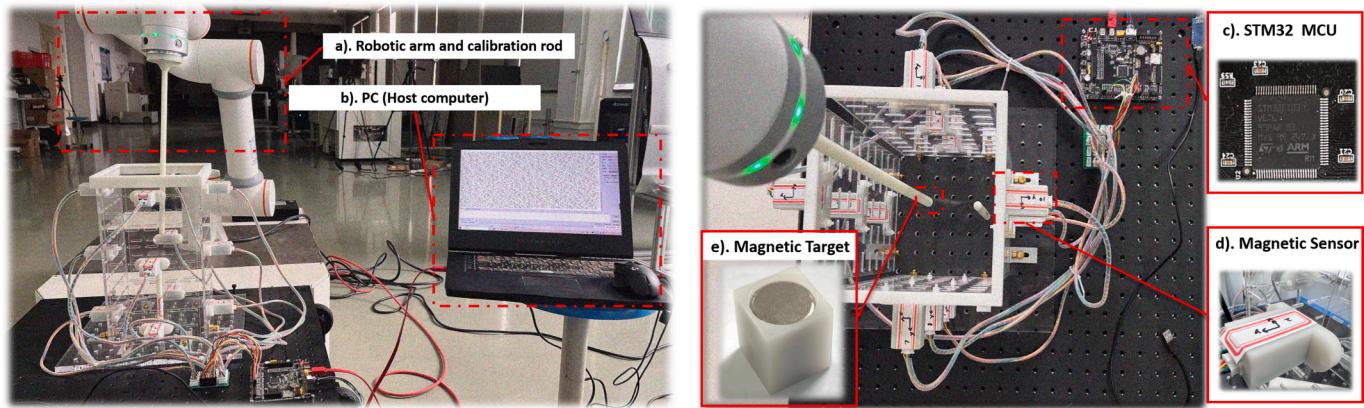


Fig. 4. Experimental Platform. Including: a) Robotic Arm and Calibration Rod, b)PC(Host Computer), c) STM32 MCU, d) Magnetic Sensor, and e) Magnetic Target.

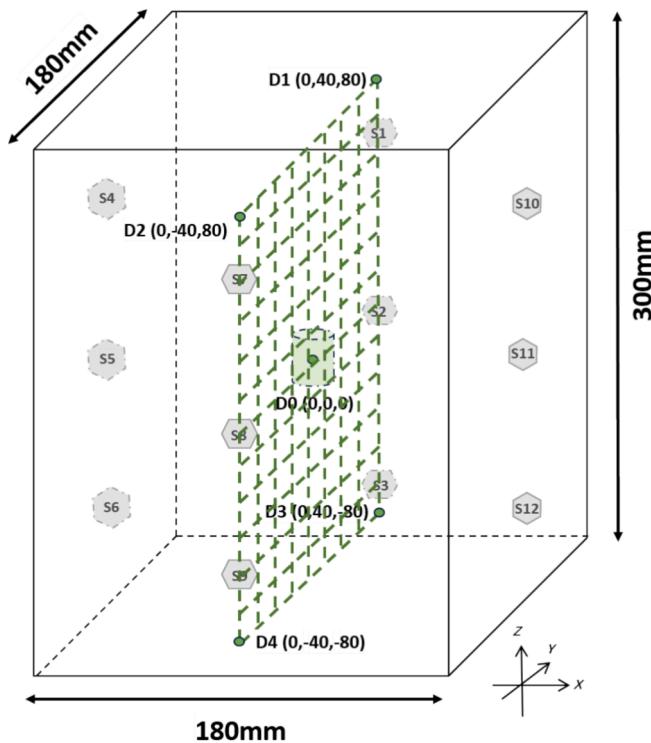


Fig. 5. Position of the experimental plane within the overall experimental platform space, with sampling points evenly distributed within the plane.

obtained by the MP-KAN model, along with those of other models, and evaluates these results using a variety of evaluation metrics. In the concluding section, we discuss the advantages, limitations, and potential future directions of the MP-KAN model. Finally, we summarize our findings in this section.

The traditional MLP model is typically illustrated in Fig. 2. It generally consists of a fully connected structure with three main components: an input layer, hidden layers, and an output layer [24,25]. The model functions by setting a fixed activation function at the neuron nodes and placing learnable weights on the “edges”. A general L-layer MLP can be expressed as:

$$\text{MLP}(\mathbf{x}) = (\mathbf{W}_{L-1} \circ \sigma \circ \mathbf{W}_{L-2} \circ \dots \circ \mathbf{W}_1 \circ \sigma \circ \mathbf{W}_0) \mathbf{x} \quad (1)$$

However, KAN places learnable activation functions on the “edges,” with the weight parameters replaced by learnable one-dimensional functions that are parameterized as spline functions. At the nodes,

only summation operations are performed. This allows KAN to achieve superior performance in physical problems such as magnetic positioning, which rely on structured data and require precise data fitting capabilities. The Kolmogorov-Arnold Superposition Theorem (KART) is typically expressed as [26]:

$$f(\mathbf{x}) = f(x_1, \dots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^n \phi_{q,p}(x_p) \right) \quad (2)$$

This indicates that when $f(\cdot)$ is a multivariate continuous function on a bounded domain, it can be expressed as a two-layer nested superposition of a finite number of univariate continuous functions. Furthermore, the theorem suggests that the summation operation can be utilized to achieve multivariate function fitting. In the case of magnetic positioning problems, magnetic field intensity information can often be fitted using multivariate polynomial functions (which are typically continuous and smooth) [10]. This gives the KAN architecture an opportunity to perform better on such problems.

KAN has the same fully connected structure as MLP, consisting of an input layer, hidden layers, and an output layer. For the construction of MP-KAN, the first step is to collect the original spatial three-component magnetic field intensity through a magnetic sensor array and convert it into textual information. In the second step, the data text is processed on an upper computer to extract sensor data and construct an overall feature sequence of the sensors [Mag0X, Mag0Y, Mag0Z ... MagnZ] as the input layer of the model. Here, the parameter n represents the magnetic field intensity of the n th sensor in the corresponding spatial direction. Since the experiments in this paper are conducted on the plane where $X = 0$, with the X position fixed, we ultimately select the magnet position $[Y, Z]$ as the prediction sequence (output layer). Below, we will introduce the training process of MP-KAN in detail.

1). Construction of Training Data: Utilize the upper computer to extract the target data from the text files, and then reconstruct the data into a $[1 \times L]$ structure, where L represents the total number of sensor channels and magnet positions, totaling 38 in this paper. By conducting N experiments for each point within the plane, a data matrix of size $[(153 \times N) \times L]$ can ultimately be obtained as the training data for the MP-KAN.

2). Structure of MP-KAN: Its structure is shown in Fig. 3. The model places learnable activation functions on the “edges,” and the activation function $\phi(x)$ is set as shown in (3):

$$\phi(x) = f(x) + B_spline(x) \quad (3)$$

Among them, there are sometimes issues of gradient vanishing in the data collected by the magnetic positioning system, as well as the problem of zero gradient for negative values when selecting the relu function as the activation function. Therefore, the silu function is chosen as the basis function $f(x)$, as shown in equation (4). This not only ensures the

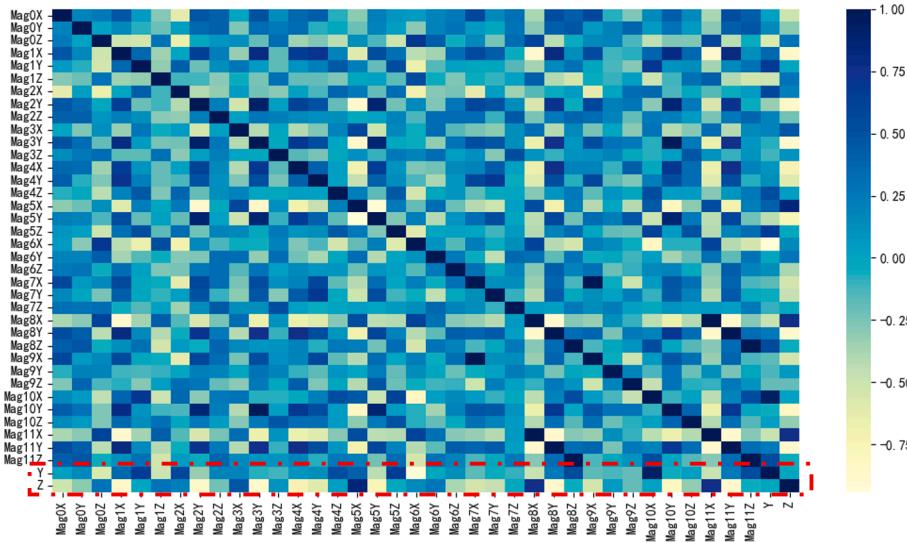


Fig. 6. Spearman's Rank Correlation Coefficient Matrix between Magnetic Field Intensity of Each Channel and Spatial Position.

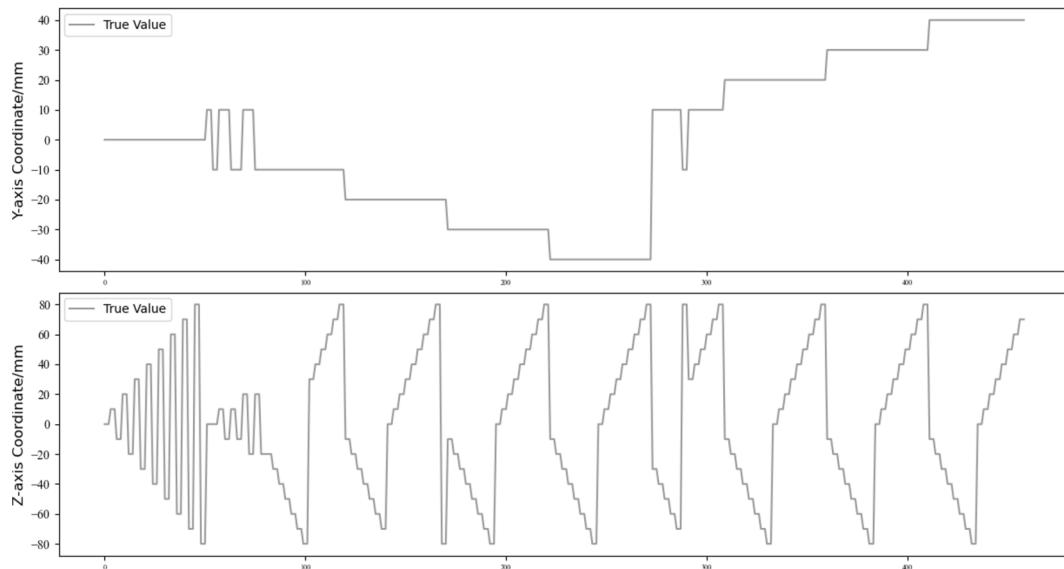


Fig. 7. Actual Position Coordinates. The upper part shows the Y-axis coordinates; The lower part shows the Z-axis coordinates.

Table 1
Test results of magnetic positioning models based on different machine learning algorithms.

Models	Y-axis			
	MAE	MSE	RMSE	R ²
MP-RF	0.12935	0.03927	0.19816	0.99372
MP-XGBoost	0.16206	0.04820	0.21953	0.99230
MP-NB	1.73913	4.60870	2.14679	0.26332
MP-DT	0.08696	0.08696	0.29488	0.98610
Models	Z-axis			
	MAE	MSE	RMSE	R ²
MP-RF	0.21087	0.50788	0.71266	0.97678
MP-XGBoost	0.22299	0.30529	0.55253	0.98604
MP-NB	0.82609	1.80000	1.34164	0.91772
MP-DT	0.03478	0.03478	0.18650	0.99841

nonlinearity and continuous differentiability of the activation function, but also allows the activation function to be defined within the range from negative infinity to positive infinity.

$$f(x) = \text{silu}(x) = \frac{x}{1 + e^{-x}} \quad (4)$$

$B_{\text{spline}}(x)$ is a spline function that fits a curve through segmented continuous piecewise curves. During training, each activation function is initialized as $B_{\text{spline}}(x) \approx 0$, and b_i serves as a trainable parameter to ensure that the activation function is learnable.

$$B_{\text{spline}}(x) = \sum_i b_i B_i(x) \quad (5)$$

Based on the input activation functions, the grid is updated in real-time every twenty training steps, with a grid interval set to 3 ($G = 3$). Finally, at the nodes, for an $l+1$ layer network structure, the activation value of a neuron will be the sum of all incoming post-activation values. A general l -layer KAN network can be expressed as:

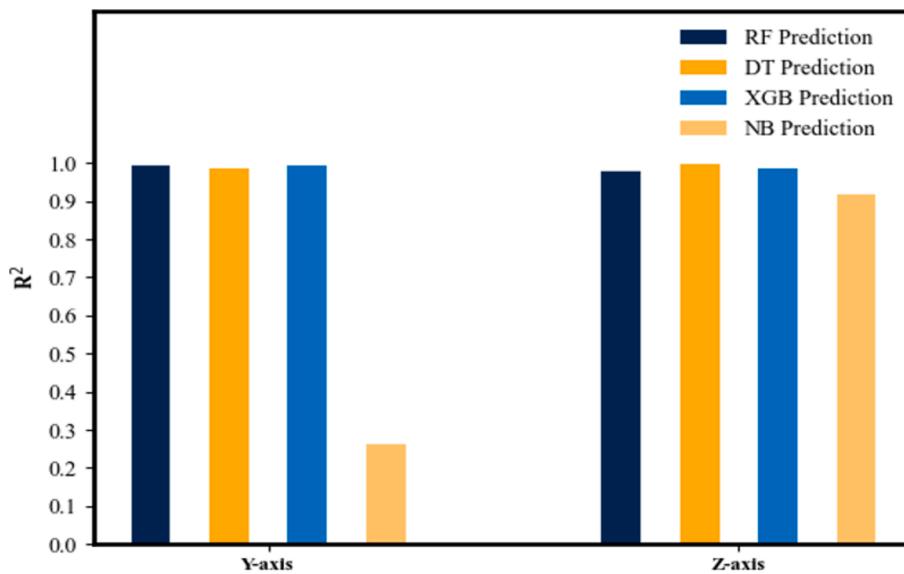


Fig. 8. Comparison of R^2 Values between machine learning models for Y-axis and Z-axis Prediction Results.

$$\text{KAN}(x) = (\Phi_{L-1} \circ \Phi_{L-2} \circ \dots \circ \Phi_1 \circ \Phi_0)x \quad (6)$$

3). Simplification of the MP-KAN Prediction Model: To achieve model sparsity, the L1 norm of the activation functions is adopted, and additional entropy regularization is incorporated. Combining the structure of the input and output layers, we define the L1 norm of Φ as the sum of the L1 norms of all activation functions, and its entropy as $S(\Phi)$, expressed as (7) and (8) respectively.

$$|\Phi|_1 \equiv \sum_{i=1}^{38} \sum_{j=1}^2 |\phi_{ij}|_1 \quad (7)$$

$$S(\Phi) \equiv - \sum_{i=1}^{38} \sum_{j=1}^2 \frac{|\phi_{ij}|_1}{|\Phi|_1} \log \left(\frac{|\phi_{ij}|_1}{|\Phi|_1} \right) \quad (8)$$

The total training objective L_{total} is the prediction loss L_{pred} plus the L1 and entropy regularization for KAN layers.

$$L_{\text{total}} = L_{\text{pred}} + \lambda \left(\sum_{L=0}^{L-1} |\Phi_L|_1 + \sum_{L=0}^{L-1} S(\Phi_L) \right) \quad (9)$$

Here, L_{pred} represents the Root Mean Square Error (RMSE) between the predicted position and the actual position, λ controls the overall regularization strength, L represents the Model depth. Additionally, after training with the sparsity penalty, we prune the unimportant neurons at the nodes by setting a threshold parameter $\theta = 0.01$, achieving optimal regression prediction performance. Finally, the specific parameter settings for the models are as follows:

Training Setting of MP-KAN: The model parameters were determined primarily through empirical and manual tuning based on the changes in the training loss curve. In the three-layer structure of MP-KAN, the number of neurons in the hidden layer was set to 5, while the input layer and output layer were set to 30 and 2 respectively. The number of spline curves was set to 3, and the grid interval was set to 3. The LBFGS optimizer was used for training over 100 epochs.

Training Setting of MP-MLP: Combining the experience of manual parameter tuning from [27], a three-layer structure was ultimately selected, with the number of neurons in the hidden layer set to 5. The ReLU function was used as the activation function, and the RMSE was set as the loss function. The Adam optimizer was used for training over 400 epochs, with all other settings kept as default to achieve optimal performance.

Training Setting of Traditional Machine Learning Algorithms: All machine learning algorithms utilized the grid search method [28] to optimize hyperparameter settings. For example, in the random forest algorithm, the number of decision trees was set to 8, and all other settings were kept as default to achieve optimal performance.

3. Experimental and evaluation indicators

3.1. Experiment setup

The overall structure of the experimental platform is shown in the Fig. 4. The experimental platform has been meticulously designed, featuring an overall framework constructed from acrylic material, shaped into a cubic form with dimensions of 300 mm \times 180 mm \times 180 mm. This configuration aims to emulate the positioning space encountered in practical application scenarios, offering a realistic simulation environment. A neodymium-iron-boron permanent magnet, magnetized specifically along the positive Z-axis direction, serves as the magnetic target. This magnet boasts a diameter (Φ) and height (H) of 10 mm each, ensuring a consistent and reliable magnetic field. To maintain the magnet's spatial attitude during movement and prevent any disruptions to positioning accuracy, it is securely connected to a robotic arm via a calibration rod. The robotic arm plays a crucial role, providing the necessary driving and control functionalities to guarantee precise movement positions of the magnet [29,30]. For the sensor array, a tri-axes digital magnetic sensor, namely the LIS3MDL, has been selected. This sensor offers the flexibility to arrange any number of units in any position on the experimental framework, catering to various experimental requirements. In this study, 12 sensors have been uniformly distributed across four planes, ensuring comprehensive and accurate data collection. Data acquisition from the sensor array is facilitated through I2C communication with an STM32 microcontroller unit (MCU). This setup allows for seamless data transfer and processing. Finally, the experimental data is transmitted to a personal computer (PC) in the form of text files, utilizing serial communication. Except for the embedded system components, which were programmed in language C, all other programs were developed using Python 3. Experimental equipment features the following specifications: CPU-Intel Core i7-7700HQ, GPU-NVIDIA GTX 1060, RAM- 8 Gibbs.

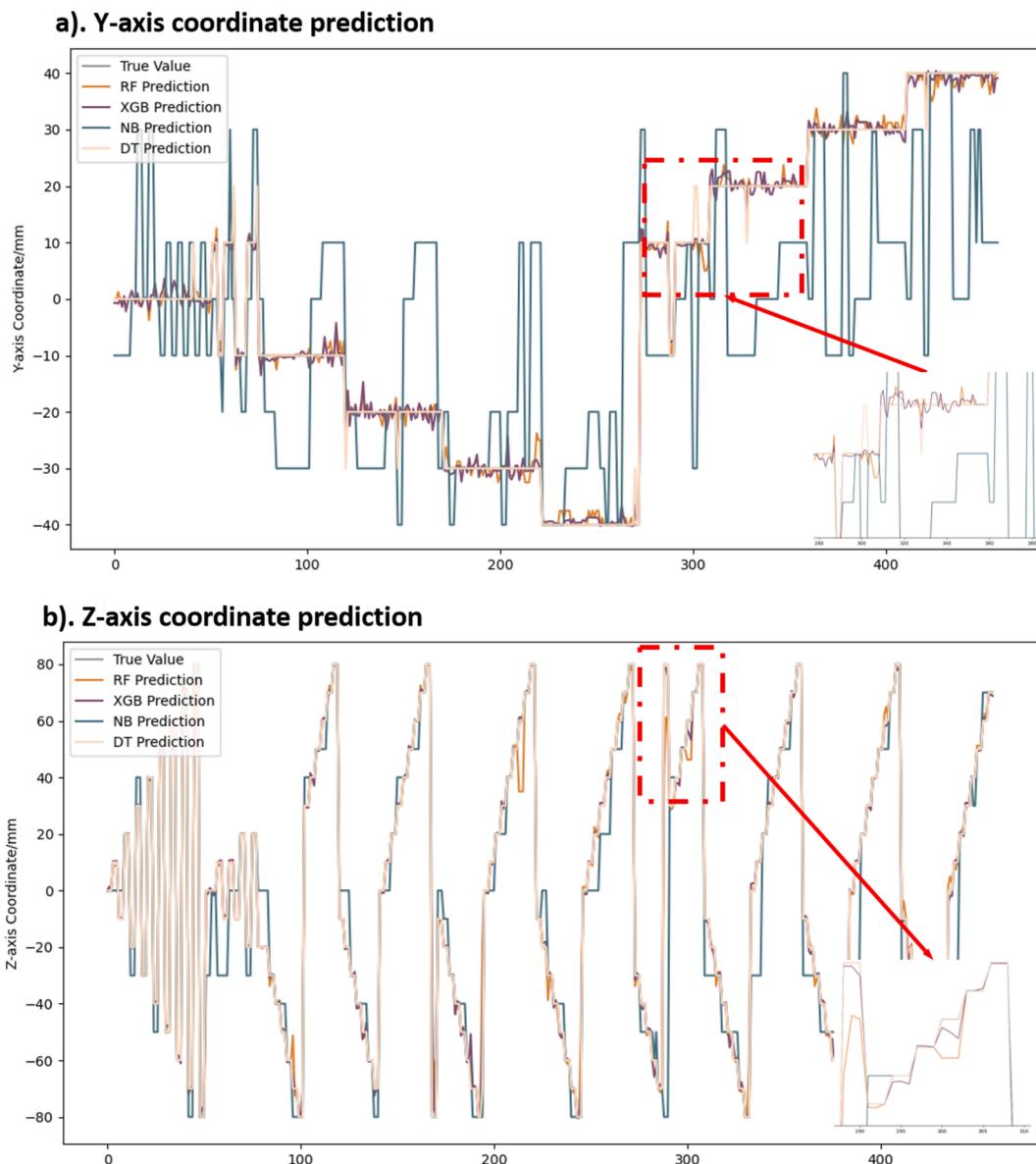


Fig. 9. Regression Prediction Results of Machine Learning Models. a) Y-axis Coordinate Prediction. b) Z-axis Coordinate Prediction.

Table 2
Test results of MP-MLP and MP-KAN.

Evaluation metrics	MP-MLP		MP-KAN	
	Y-axis	Z-axis	Y-axis	Z-axis
MAE	0.19376	0.27210	0.07638	0.10187
MSE	0.06898	0.24026	0.00981	0.02345
RMSE	0.26264	0.49016	0.09906	0.15314
R ²	0.98897	0.98902	0.99843	0.99893

3.2. Data preprocessing

The experiment focuses on a single plane within the space, measuring 80 mm × 160 mm, as shown in Fig. 5. On the measurement plane formed by D1, D2, D3, and D4, data collection is performed at points spaced 10 mm apart horizontally and vertically. Each point is sampled three times, and the magnetic field intensity signals of each channel are normalized using min–max scaling. The formula as follow:

$$x_{\text{normalization}} = \frac{x - \min(X)}{\max(X) - \min(X)} \quad (10)$$

Regarding data partitioning, the entire dataset is divided into four parts, with the first three parts used for model training and the last part for model testing. This method effectively validates the robustness of the model.

3.3. Evaluation metrics

In this article, we conduct a direct comparison between predicted data and actual data to visually illustrate the predictive accuracy of various models. Furthermore, we utilize quantitative evaluation metrics, including the coefficient of determination (R²), mean absolute error (MAE), and root mean square error (RMSE), to rigorously assess the regression performance of the models. The formulas for these metrics are given as follows [28,31]:

$$R^2 = 1 - \frac{\sum_i (\hat{y}^{(i)} - y^{(i)})^2}{\sum_i (\bar{y} - y^{(i)})^2} \quad (11)$$

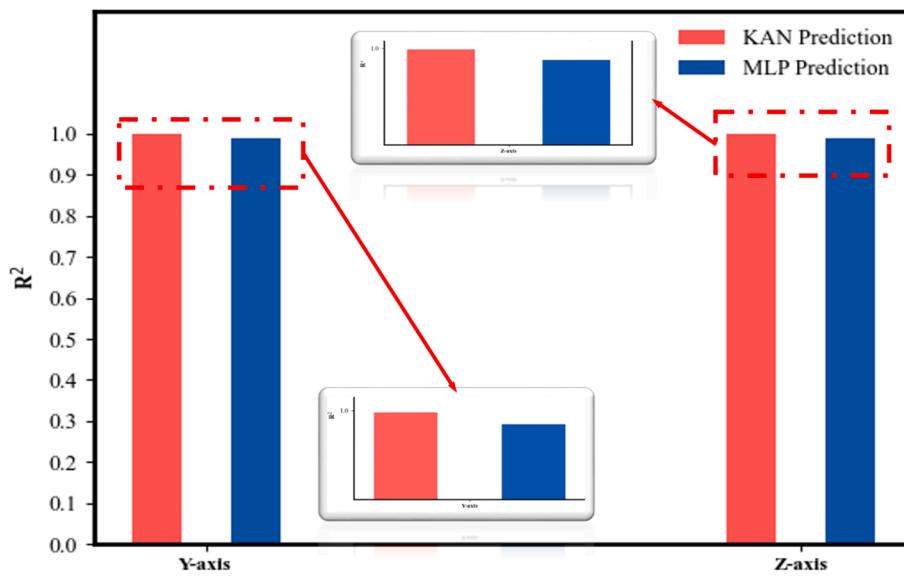


Fig. 10. Comparison of R^2 Values between Two Neural Network Models for Y-axis and Z-axis Prediction Results.

$$\text{RMSE} = \sqrt{\frac{1}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})^2} \quad (12)$$

$$\text{MAE} = \frac{1}{m} \sum_{i=1}^m |y^{(i)} - \hat{y}^{(i)}| \quad (13)$$

As seen in formula (11)–(13), the MAE directly reflects the true error of the model, and a smaller value indicates that the model's predictions are closer to the true values. The numerator of R^2 is the sum of squared differences between the predicted values $\hat{y}^{(i)}$ and the true values $y^{(i)}$ which is known as the mean squared error (MSE), while the denominator is the sum of squared differences between the mean true value \bar{y} and the true values $y^{(i)}$, which is the variance. Compared to MAE and RMSE, R^2 constrains the regression results within the range of 0 to 1, making it convenient for us to compare the performance of different models. Additionally, R^2 exhibits greater robustness to the error range of predicted data. When the model achieves perfect estimation, $R^2 = 1$, and the closer R^2 is to 1, the better the model performs.

4. Experimental result

4.1. Feature analysis and data preprocessing

The rank correlation coefficient is often used to reflect the correlation between the direction and strength of the changing trends of two random variables. We can employ it as a feature engineering technique to check the feasibility of using input data for model training [32]. Since the training data in this paper exhibits nonlinear relationships, we select the Spearman rank correlation coefficient to reflect the strength of the monotonic relationship between each sensor channel and the magnetic target position. Its formula can be simply expressed as (13).

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \quad (13)$$

where d_i represents the difference in rank values of the i -th data pair, and n is the total number of observation samples. As shown in Fig. 6, we visualize the Spearman rank correlation coefficients between the magnetic field intensity of each channel and the predicted spatial position. It can be seen that for the predicted spatial position, the magnetic sensor data generally exhibits moderate or strong correlation ($R_s > 0.25$) or ($R_s < -0.25$). Because of this, in this article, data collected from all channels

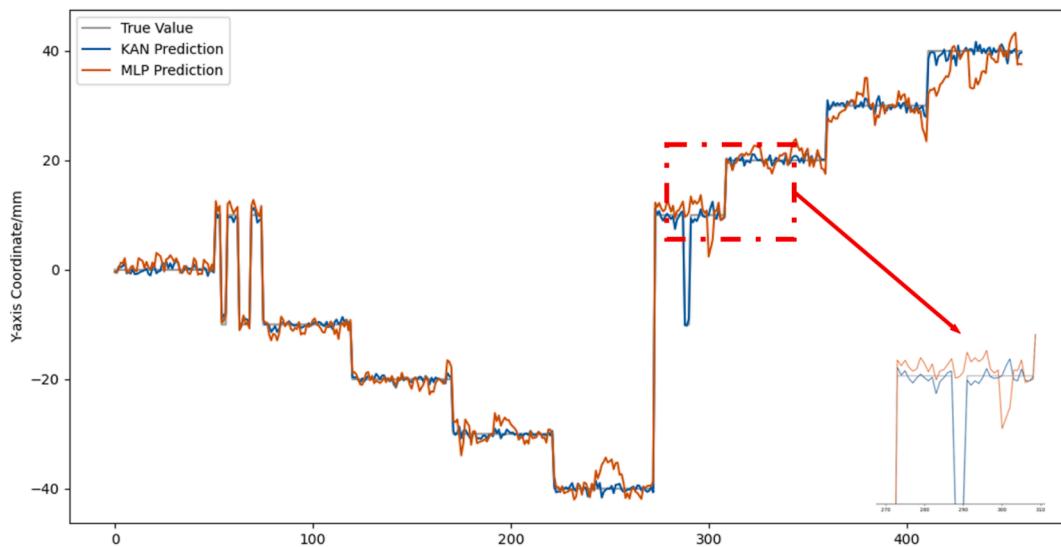
can be used as inputs for training the model.

Data shuffling as a preprocessing method that does not change the data itself but changes the order of the data to prevent the model from learning the sequential pattern of the data. We introduce this method to avoid overfitting of the model to a certain extent. Therefore, in order to further observe the prediction results intuitively, we shuffled the input data, and its distribution curve is shown in Fig. 7.

4.2. Comparison of experimental results of traditional machine learning algorithm

To improve the prediction accuracy in multi-output regression problems, traditional machine learning algorithms often utilize wrapper methods. These approaches can be classified into two types: individual models for each output (MultiOutputRegressor) and chained models for each output (RegressorChain). Among them, the RegressorChain model leverages the input to predict an output using the first model, and then the second model utilizes both the input and the output from the first model to predict the next output, and so on, to obtain all predictions. However, this approach can affect the independence between output variables. Therefore, this paper selects MultiOutputRegressor to wrap various models, selects two classic models from ensemble machine learning and basic machine learning models for training, and inputs the test set data into the trained models. The data presented in Table 1 showcases the evaluation metrics of various machine learning algorithms. Notably, the Naive Bayes (NB) algorithm exhibits the highest MAE and RMSE, indicating its difficulty in enhancing positioning accuracy when faced with magnetic localization tasks. Random Forest (RF) and Extreme Gradient Boosting (XGBoost) algorithm, both ensemble machine learning algorithms, significantly improve the precision of experimental results compared to the NB algorithm. In Fig. 8, it can be seen that RF achieved R^2 of 0.99372 on the Y-axis and 0.97678 on the Z-axis, while achieved R^2 of 0.99230 on the Y-axis and 0.98604 on the Z-axis. Meanwhile, both ensemble algorithms have lower MAE and RMSE compared to the NB algorithm. However, based solely on the MAE, the Decision Tree (DT) algorithm seems to have more accurate results than the ensemble machine learning algorithms. However, since all the data points in this experimental dataset are integer values, the DT algorithm produced identical MAE and MSE values, which is unlikely to occur with a larger dataset and denser test points. Therefore, the actual performance of the DT model will not be as significant as the evaluation metrics indicate [16]. As shown in Fig. 9, comparing it with other

a). Y-axis coordinate prediction



b). Z-axis coordinate prediction

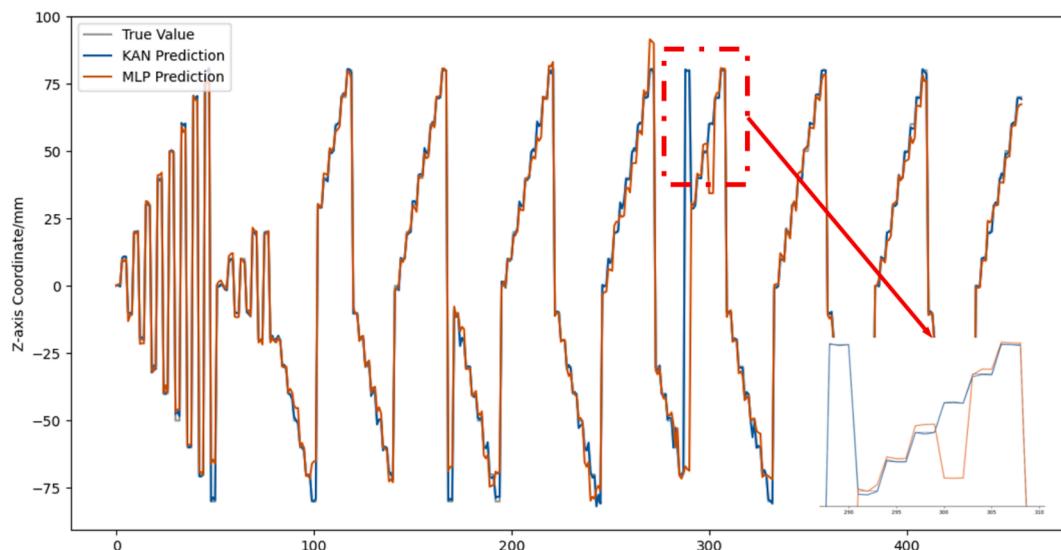


Fig. 11. Neural Network Regression Prediction Results. a) Y-axis Coordinate Prediction. b) Z-axis Coordinate Prediction.

Table 3

Comparison with existing magnetic positioning methods.

Parameters	This work	[6]	[10]	[11]	[16]	[19]
Positioning method	Neural network	Differential signal	Magnetic dipoles	Optimization algorithm	Machine learning	Neural network
Near-field error	No	Yes	Yes	NR	Yes	No
Multi-axial prediction results	Balanced	Balanced	Unbalanced	NR	NR	Unbalanced
Sensor layout	Space multi-plane	Single plane	Space multi-plane	Single plane	Single plane	Single plane
Attitude prediction	No	Yes	Yes	No	No	No
MAE	0.10187 mm	NR	NR	NR	NR	NR
RMSE	0.15314 mm	NR	NR	NR	NR	0.01 mm
Maximum positioning error	0.24 mm	5.2 mm	3.82 mm	5.30 mm	<0.1 cm	0.396 mm

Note: NR: No reported.

models reveals that the DT predictions tend to favor integer values rather than the required floating-point values. In the labeled area, it can also show that RF and XGBoost algorithms have fewer spikes, and more spikes in the location map often represent poor stability of this positioning algorithm, combined with RMSE indicators that are more sensitive to outliers, we will find out that integrated machine learning algorithms perform better. This case of DT algorithm reveals that the

evaluation metrics of the magnetic positioning system must be analyzed in conjunction with coordinate plots to accurately reflect the true performance of the algorithm. Ultimately, the results indicate that integrated machine learning models can provide more precise predictions; however, they still suffer from an imbalance in prediction outcomes along the Y-axis and Z-axis. These models often maintain good prediction results in a single direction only. For instance, while the Random

Forest (RF) algorithm achieves an RMSE of 0.19816 on the Y-axis, this value triples on the Z-axis, suggesting limited generalization capability of the model. Therefore, a more balanced positioning algorithm is required.

4.3. Comparison of neural network model experimental results

By introducing hidden layers, neural networks can learn and characterize more complex nonlinear relationships. This capability enables neural networks to handle more complex multi-output problems than machine learning algorithms. The data in Table 2 reflects the evaluation metrics results of the MP-KAN and MP-MLP on the data set. It is evident that, compared to models built using machine learning algorithms, MP-MLP and MP-KAN exhibit more balanced positioning accuracy along the Y-axis and Z-axis, with differences in R^2 not exceeding 0.1 %. Notably, for the MP-KAN model, the differences in MAE and RMSE between the two axial directions have been narrowed to 2.549e-2 and 5.408e-2, respectively. Furthermore, this model not only produces more balanced prediction results but also demonstrates R^2 values greater than 0.99 along both the Y-axis and Z-axis, as shown in Fig. 10. This represents the optimal outcome among all models studied in this paper, achieving high-precision fitting of the positional curves. Combining a more intuitive representation method, in the labeled area of the Fig. 11, since we shuffled the original data, the original curve image exhibits significant positional changes in the range of points 250 to 300. In comparison to MP-MLP, MP-KAN clearly performs better at predicting these positions, and this advantage will become even more prominent with a larger training set. Then, compared to the RMSE value, on the Y-axis and Z-axis, they have been reduced to 62.28 % and 68.76 % of the value of else models.

Further compared the work, the comparison with the existing magnetic positioning methods is shown in Table 3. It can be inferred that the method proposed in this paper achieves high-precision spatial localization. Compared to the approaches in [6] and [11], it demonstrates superior localization accuracy under conditions where the sensor distribution aligns more closely with practical application scenarios. In contrast to [10], with the same sensor distribution, it resolves the issue of uneven localization errors in multi-axis scenarios. When compared with [16], this paper further discusses magnetic localization algorithms based on machine learning and presents more detailed evaluation metric results, concluding that neural networks can achieve higher localization accuracy than traditional machine learning methods. In comparison with [19], a different neural network algorithm is employed, reducing the maximum localization error by 0.15 mm.

5. Conclusion

This article proposes a magnetic localization model based on the Kolmogorov-Arnold network. The objective of this approach is to address the inaccuracies in near-field predictions and the uneven errors in spatial position predictions of magnetic targets associated with traditional magnetic localization methods, while simultaneously enhancing the system's localization accuracy. By utilizing a three-layer network structure, we have achieved balanced evaluation metrics for the system along the y-axis and z-axis, with Y_{MAE} , Y_{MSE} , and Y_{R2} reaching 0.07638, 0.00981, 0.99843; and Z_{MAE} , Z_{MSE} , and Z_{R2} reaching 0.10187, 0.02345, and 0.99893. Experimental outcomes demonstrate that the relative positional error can be reduced to 0.25 % of the magnetic field intensity at the point, with a maximum localization error of 0.24 mm, achieving both uniformity and high precision. This paper innovatively introduces neural networks into the magnetic localization system, providing an efficient solution for magnetic localization methods within spatial domains. In subsequent work, we will further expand the detection range of the magnetic localization system and explore the potential applications of neural network deployment in magnetic localization systems.

CRediT authorship contribution statement

Zibo Gao: Writing – original draft, Software, Methodology, Investigation, Data curation. **Ming Kong:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Funding acquisition, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

- [1] G. Iddan, G. Meron, A. Glukhovsky, P. Swain, Wireless capsule endoscopy, *Nature* **405** (2000) 417–418.
- [2] P. Valdastri, M. Simi, R.J. Webster, Advanced technologies for gastrointestinal endoscopy, *Annu. Rev. Biomed. Eng.* **14** (2012) 397, <https://doi.org/10.1146/annurev-bioeng-071811-150006>.
- [3] B. Park, H. Myung, Underground localization using dual magnetic field sequence measurement and pose graph SLAM for directional drilling, *Meas. Sci. Technol.*, **25** (12) (Dec. 2014) 125101, doi: [10.1088/0957-0233/25/12/125101](https://doi.org/10.1088/0957-0233/25/12/125101).
- [4] Z. Liu, et al., Drilling localization and error analysis of radial horizontal jet drilling based on magnetic gradient tensor, *Energies* **13** (19) (Sep. 2020) 4989, <https://doi.org/10.3390/en13194989>.
- [5] A. Kuwahata, et al., Development of magnetic probe for sentinel lymphnode detection in laparoscopic navigation for gastric cancer patients, *Sci. Rep.* **10** (1) (Feb. 2020) 1798, <https://doi.org/10.1038/s41598-020-58530-5>.
- [6] S. Song, S. Wang, S. Yuan, J. Wang, W. Liu, M.Q.H. Meng, Magnetic tracking of wireless capsule endoscope in mobile setup based on differential signals, *IEEE Trans. Instrum. Measure.* **70** (2021) 1–8, <https://doi.org/10.1109/tim.2021.3069488>.
- [7] D. Pan et al., Avoidance Method of Magnetic Target Positioning-Inaccuracy-Area, *IEEE Trans. Instrum. Measure.*, **73** (2024) 1–9, Art no. 6003409, doi: [10.1109/TIM.2023.3267371](https://doi.org/10.1109/TIM.2023.3267371).
- [8] C. Hu, M.-Q.-H. Meng, M. Mandal, Efficient linear algorithm for magnetic localization and orientation in capsule endoscopy, in: Proc. IEEE Eng. Med. Biol., Jan. 2005, pp. 7143–7146.
- [9] C. Hu, M.-Q.-H. Meng, M. Mandal, A linear algorithm for tracing magnet position and orientation by using three-axis magnetic sensors, *IEEE Trans. Magn.* **43** (12) (Dec. 2007) 4096–4101.
- [10] C. Hu, et al., Locating intra-body capsule object by three-magnet sensing system, *IEEE Sensors J.* **16** (13) (2016) 5167–5176.
- [11] B. Ye et al., A novel magnetic spiral capsule endoscope localization method based on an improved artificial bee colony algorithm, *IEEE Sensors J.*, **24**(2) (15 Jan. 15, 2024) 1740–1750, doi: [10.1109/JSEN.2023.3336167](https://doi.org/10.1109/JSEN.2023.3336167).
- [12] Q. Fu, C. Fan, C. Wang, S. Zhang, Z. Xi, Improved Magnetic Positioning of Medical Capsule Robot, *IEEE Sens. J.*, **23**(15) (1 Aug. 1, 2023) 17391–17398, doi: [10.1109/JSEN.2023.3284911](https://doi.org/10.1109/JSEN.2023.3284911).
- [13] W. Song et al., Magnetic sensor array based on coordinate measuring and differential evolution algorithm, *IEEE Sens. J.*, **23**(17) (1 Sept. 1, 2023) 19137–19144, doi: [10.1109/JSEN.2023.3287589](https://doi.org/10.1109/JSEN.2023.3287589).
- [14] S. Lin, et al., Improvement and omnidirectional analysis of magnetic gradient tensor invariants method, *IEEE Trans. Ind. Electron.* **68** (8) (Aug. 2021) 7603–7612, <https://doi.org/10.1109/TIE.2020.3001844>.
- [15] X. Guo, et al., Pose tracking method using magnetic excitations with frequency division for robotic endoscopic capsules, *Biomed. Microdevices* **24** (1) (2022) 9.
- [16] L. Nguyen-Ngoc, Q. Nguyen-Huu, G. De Roeck, T. Bui-Tien, M. Abdel-Wahab, Deep neural network and evolved optimization algorithm for damage assessment in a truss bridge, *Mathematics* **12** (2024) 2300, <https://doi.org/10.3390/math12152300>.
- [17] YiFei Li, Hoang-Le Minh, MaoSen Cao, Xiangdong Qian, Magd Abdel Wahab, An integrated surrogate model-driven and improved termite life cycle optimizer for damage identification in dams, *Mech. Syst. Signal Process.*, **208** (2024) 110986, ISSN 0888-3270, Doi: [10.1016/j.ymssp.2023.110986](https://doi.org/10.1016/j.ymssp.2023.110986).
- [18] Van-Thien Tran, Trung-Kien Nguyen, H. Nguyen-Xuan, Magd Abdel Wahab, Vibration and buckling optimization of functionally graded porous microplates using BCMO-ANN algorithm, *Thin-Walled Structures*, **182**(Part B) (2023) 110267, ISSN 0263-8231, Doi: [10.1016/j.tws.2022.110267](https://doi.org/10.1016/j.tws.2022.110267).
- [19] A.-I. Sasaki, E. Ohta, Magnetic-Field-Based Position Sensing Using Machine Learning, *IEEE Sens. J.*, **20**(13) (1 July 1, 2020) 7292–7302, doi: [10.1109/JSEN.2020.2979071](https://doi.org/10.1109/JSEN.2020.2979071).
- [20] W. He, K. Xu, G. He, A Method of Ferromagnetic Object Identification Based on Magnetic Sensor and Machine Learning, in: 2023 IEEE 16th International

- Conference on Electronic Measurement & Instruments (ICEMI), Harbin, China, 2023, pp. 446-451, doi: [10.1109/ICEMI59194.2023.10270466](https://doi.org/10.1109/ICEMI59194.2023.10270466).
- [21] B. Lv, Y. Chen, H. Dai, S. Su, M. Lin, PKBPNN-based tracking range extending approach for TMR magnetic tracking system, *IEEE Access* 7 (2019) 63123–63132, <https://doi.org/10.1109/ACCESS.2019.2917140>.
- [22] Q. Fu, D. Zhao, L. Shao, S. Zhang, Magnetic Localization Algorithm of Capsule Robot Based on BP Neural Network, *IEEE Trans. Instrum. Measure.*, 73 (2024) 1-9, Art no. 4002509, doi: [10.1109/TIM.2023.3341129](https://doi.org/10.1109/TIM.2023.3341129).
- [23] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljavić, T.Y. Hou, M. Tegmark, KAN:Kolmogorov-Arnold Networks, 2024, <https://api.semanticscholar.org/CorpusID:269457619>.
- [24] X. Chen, et al., A novel artificial neuron-like gas sensor constructed from CuS quantum dots/Bi₂S₃ nanosheets, *Nano-Micro Lett.* 14 (8) (Dec. 2022) 1–15.
- [25] Y. Cui et al., Investigation on the ignition delay prediction model of multi-component surrogates based on back propagation (BP) neural network, *Combustion Flame*, 237 (Mar. 2022) Art. no. 111852.
- [26] A.N. Kolmogorov, V.I. Arnold, On the representation of continuous functions of several variables by superpositions of continuous functions of one variable and addition, *Doklady Akademii Nauk SSSR* 114 (5) (1957) 953–956.
- [27] T. Bao, S.A.R. Zaidi, S. Xie, P. Yang, Z.-Q. Zhang, A CNN-LSTM Hybrid Model for Wrist Kinematics Estimation Using Surface Electromyography, *IEEE Trans.* Instrum. Measure., 70 (2021) 1-9, Art no. 2503809, doi: [10.1109/TIM.2020.3036654](https://doi.org/10.1109/TIM.2020.3036654).
- [28] S. Chen, W. Zhang, R. Luo, Y. Zhao, Y. Yang, B. Zhang, B. Hu, System energy and band gap prediction of titanium dioxide based on machine learning, *J. Mol. Struct.* 1307 (2024), <https://doi.org/10.1016/j.molstruc.2024.137934>.
- [29] L. Mao, S. Shuang, H. Chao, C. Dongmei, M.Q. Meng, A novel method of 6-Dof electromagnetic navigation system for surgical robot, in: Proc. 8th World Congr. Intell. Control Automat.Jul. 2010, pp. 2163–2167, doi: [10.1109/WCICA.2010.5554348](https://doi.org/10.1109/WCICA.2010.5554348).
- [30] W. Yang, C. Hu, M.-Q.-H. Meng, S. Song, H. Dai, A six-dimensional magnetic localization algorithm for a rectangular-magnet objective based on a particle swarm optimizer, *Ieeetrans. Magn.* 45 (8) (Aug. 2009) 3092–3099, <https://doi.org/10.1109/TMAG.2009.2019116>.
- [31] A. Alotaibi, T. Gul, I.M. Saleh Alotaibi, A. Alghuried, A.S. Alshomrani, M. Alghuson, Artificial neural network analysis of the flow of nanofluids in a variable porous gap between two inclined cylinders for solar applications, *Eng. Appl. Computational Fluid Mech.* 18 (1) (2024), <https://doi.org/10.1080/19942060.2024.2343418>.
- [32] S.Y. Li, E.A. Kronberg, C.G. Mouikis, H. Luo, Y.S. Ge, A.M. Du, Prediction of proton pressure in the outer part of the inner magnetosphere using machine learning, *Space Weather* 21 (9) (2023), <https://doi.org/10.1029/2022sw003387>.