Detection and recognition of mines using magnetic field sensors and Kolmogorov-Arnold networks

Roman Peleshchak[1,†], Vasyl Lytvyn[1,†], Ivan Peleshchak[1,*,†], Viacheslav Beltiukov[1,*,†], Danylo Chystyakov[2,†] and Yaroslav Kis[1,†]

[1] *Lviv Polytechnic National University, 12 Bandera Street, 79013, Lviv, Ukraine*
[2] *National Forestry University of Ukraine, 103 Gen. Chuprynky St., 79057, Lviv, Ukraine*

Abstract

This paper presents a passive demining method that integrates fluxgate magnetic sensing with Kolmogorov–Arnold Networks (KANs) for multiclass landmine recognition. Magnetic-field anomalies recorded by an FLC-100 sensor above buried targets are combined with sensor height and categorical soil descriptors to form a three-dimensional feature space. To overcome data scarcity, 338 authentic measurements were augmented by injecting Gaussian noise that preserves subgroup statistics, enlarging each soil–mine pair by fifty samples and smoothing class distributions. Two spline-based architectures were compared: a compact KAN (3, 16, 16, 4) reaching accuracy of 93.56 %, and a wider KAN (3, 64, 64, 4) that offers further improvement to 95.59 % while virtually eliminating confusion between anti-personnel and booby-trap mines. Both models showed stable convergence without over-fitting, confirming the robustness of spline activations against sensor noise. Confusion-matrix analysis revealed perfect or near-perfect discrimination of "no-mine" and anti-tank cases, while remaining errors were localized to subtly differing magnetic signatures. The proposed detection is passive, avoiding the detonation risks associated with active probing and providing interpretable spline weights that expose feature contributions for safety certification. The results demonstrate the potential of physics-aware data augmentation and functional-edge neural architectures to accelerate safe demining operations.

Keywords

Kolmogorov–Arnold network, fluxgate magnetic sensor, passive mine detection, structured data

1. Introduction

Landmine detection remains a persistent and growing global concern, posing life-threatening risks to millions of people. According to the Landmine Monitor 2023, landmines and explosive remnants of war (ERW) continue to cause severe humanitarian consequences, with over 4,700 casualties reported globally in 2022 alone, the vast majority of whom were civilians. More than 60 countries remain contaminated by landmines, presenting ongoing risks for local populations, especially in post-conflict regions such as Ukraine, where land access, agricultural activity, and reconstruction are critically hindered [1]. In post-war Ukraine, the problem of landmine contamination has become especially urgent, with vast areas of agricultural and residential land requiring safe clearance. Traditional mine detection techniques often lack the reliability and responsiveness needed for large-scale humanitarian demining. Moreover, many active detection methods—based on emitting electric signals—risk triggering explosive devices, endangering human operators.

MoMLeT-2025: 7th International Workshop on Modern Machine Learning Technologies, June, 14, 2025, Lviv-Shatsk, Ukraine

* Corresponding author.
† These authors contributed equally.
roman.m.peleshchak@lpnu.ua (R. Peleshchak); vasyl.v.lytvyn@lpnu.ua (V. Lytvyn); ivan.r.peleshchak@lpnu.ua (I. Peleshchak); viacheslav.r.beltiukov@lpnu.ua (V. Beltiukov); chystyakov.d@nltu.lviv.ua (D. Chystyakov); yaroslav.p.kis@lpnu.ua (Y. Kis).
0000-0002-0536-3252 (R. Peleshchak); 0000-0002-9676-0180 (V. Lytvyn); 0000-0002-7481-8628 (I. Peleshchak); 0009-0007-1027-1540 (V. Beltiukov); 0009-0003-5968-4275 (D. Chystyakov); 0000-0002-9676-0180 (Y. Kis).

A promising alternative is the use of passive detection systems [2], particularly those based on magnetic field anomaly sensing [3, 4]. To enhance detection accuracy and reduce operational risks, modern solutions increasingly rely on machine learning techniques, including neural networks. However, neural networks often struggle with noise and distortions in real-world sensor data. One method for improving the robustness and pattern recognition capability of neural architectures is to use neural networks with embedded spline-based functional components, such as Kolmogorov–Arnold Networks (KANs), which are particularly effective at handling noisy and irregular data due to their ability to learn smooth, localized approximations of complex functions.

The aim of this study is to develop an optimized architecture of Kolmogorov-Arnold Networks in terms of the number of hidden layers, neurons per layer, and spline shape for accurate recognition of mine types in soils of varying composition. This research is highly relevant in the context of post-war recovery efforts in Ukraine, where effective and safe detection of minefields plays a crucial role in restoring civil infrastructure and ensuring public safety.

2. Literature Review

Kolmogorov–Arnold Networks (KAN) represent a recent advancement in neural network architecture inspired by the Kolmogorov–Arnold representation theorem. Unlike traditional multilayer perceptrons (MLPs) that rely on fixed activation functions at each node, KAN replaces every weight with a univariate, spline-parametrized function. This allows KANs to learn richer functional representations with fewer parameters, making them both efficient and interpretable [5].

In the work by Erdmann et al. [6], KAN was applied to a binary classification problem in high-energy physics. The authors found that while multilayer KANs did not always outperform standard MLPs in terms of accuracy, they demonstrated greater interpretability. Specifically, the activation functions learned in deeper KANs differed significantly from those in shallow models, indicating the architecture's capacity for more abstract feature extraction.

Somvanshi et al. [7] provide a comprehensive survey on KAN, outlining its theoretical foundations and practical adaptations across domains such as biomedical analytics, time series prediction, and graph learning. They highlight KAN’s flexibility and adaptability, particularly in handling high-dimensional structured data.

Barasin et al. [8] explored KAN in the context of time series classification using the UCR benchmark dataset. Their findings revealed that well-optimized KAN models outperformed MLPs and achieved competitive results compared to state-of-the-art models such as HIVE-COTE2, all while maintaining computational efficiency and robustness to hyperparameter changes.

In terms of robustness, the study published in Applied Sciences assessed the vulnerability of different KAN architectures to adversarial attacks [9]. Among the variants, KAN-Mixer showed the best performance in resisting attacks while retaining strong accuracy on clean data. This makes KAN suitable for safety-critical applications like mine detection, where robustness is paramount.

In the field of remote sensing, Cheon [10] proposed combining pretrained CNNs with KAN layers for scene classification using the EuroSAT dataset. The hybrid models achieved high classification accuracy with reduced parameter counts and faster training, illustrating the potential for integrating KAN into real-time systems.

Drokin [11] extended KAN’s application to computer vision tasks, proposing parameter-efficient KAN convolution layers and fine-tuning techniques. The results demonstrated that KAN-based models can achieve strong performance in both image classification and segmentation tasks, suggesting relevance to image-based mine detection scenarios.

The reviewed literature suggests that KAN offers a unique combination of interpretability, efficiency, and reliability across various classification domains. These characteristics make it a promising candidate for mine detection, especially in post-war Ukraine, where safety, dependability, and explainability are of paramount importance. However, the optimization of the Kolmogorov-Arnold Network architecture to improve recognition accuracy – as well as the trade-

off between training speed and recognition precision – remains an open challenge, which is crucial in the context of mine detection.

3. Classification data

In this study, we utilized the dataset provided in [4], which focuses on the classification of landmines based on magnetic field anomaly characteristics. The parameter values employed in our experimental setup are summarized in Table 1. Furthermore, we analyzed the relationship between the magnetic anomaly values and the soil type (Table 2), as well as the distance between the magnetic sensor and the buried landmine (Table 3). The general trends in magnetic field anomalies across different landmine types were also examined and illustrated (Table 3).

To obtain reliable measurements of the magnetic anomalies surrounding subsurface mines, the original study [4] employed a fluxgate magnetic sensor model FLC100 [12], which demonstrated sensitivity to minute variations in the magnetic field. This sensor-based approach enabled passive mine detection without the need for active signal emission, thus reducing the risk of accidental detonation. The design and deployment of the sensing mechanism were previously validated in [4], where a decision support system for mine classification was developed using metaheuristic classifiers.

Table 1
Input Parameters and Target Mine Classes [4]

	The Parameters			
	Input Data, "Independent Variables"			Output Data, "Dependent Variable"
	Voltage (V)	High (H)	Soil Type (S)	Mine Type (M)
Definition	The value of the output voltage of the FLC sensor due to the action of the magnetic anomaly.	The distance of the sensor above the ground.	6 different types of soil depending on the state of moisture.	Types of mines commonly found on land; 4 different classes of mines.
Limit values/Class	[0 V, 10.6 V]	[0 cm, 20 cm]	Dry and sandy	Null
			Dry and purulent	Anti-tank
			Dry and chalky	Anti-personnel
			Wet and sandy	Booby trapped Anti-personnel
			Humid and humus	
			Wet and chalky	

Table 2
Dependence of magnetic field anomalies in the vicinity of mines on soil type [4]

Soil Type	Null, VAnti-Tank, VAnti-Personnel, VBooby Trapped Anti-Personnel, V			
Dry and sandy	3.560	10.400	3.830	5.590
Dry and purulent	3.500	7.500	3.920	5.590
Dry and chalky	3.720	10.400	6.890	2.406
Wet and sandy	3.780	10.400	6.220	4.490
Humid and humus	3.350	10.400	5.050	2.770
Wet and chalky	3.610	10.400	5.960	4.400

Table 3
Dependence of the magnetic field anomaly in the vicinity of mines on the distance from the sensor to the ground [4]

Height (cm)	Mine Type 1 Voltage, V	Mine Type 2 Voltage, V	Mine Type 3 Voltage, V	Mine Type 4 Voltage, V
0.00	3.6	10.4	4.1	5.9
1.82	3.4	10.4	4.0	5.5
3.64	3.4	10.4	3.8	5.0
5.45	2.8	10.4	3.9	4.4
7.27	2.9	9.5	3.6	4.3
9.09	2.7	8.3	3.4	4.25
10.91	2.9	7.0	3.4	4.2
12.73	2.6	6.4	3.45	4.05
14.55	2.5	6.2	3.5	3.9
16.36	2.6	4.8	3.8	3.2
18.18	2.6	4.6	3.2	3.2
20.00	2.4	4.5	3.2	3.1

4. Preprocessing data

4.1. Data Generation Based on Parameterized Normal Distribution

To improve the generalization capability of the model on a limited dataset consisting of 338 real records [4], an additional data generation procedure was applied using a parameterized normal distribution.

The chosen method is based on generating new examples by adding pseudorandom noise [13] to the original feature values V and H within each subgroup of data defined by a unique pair of soil type S and mine type M. For each such subgroup, the statistical characteristics of the features are computed as follows:

$$\mu_V = \bar{V}, \sigma_V = std(V), \mu_H = \bar{H}, \sigma_H = std(H), \quad (1)$$

where:

- μ_V, μ_H — mean values of features V and H, respectively.
- σ_V, σ_H — standard deviations of features V and H.
- V, H — arithmetic means of the respective columns.
- $std(X)$ — standard deviation operator applied to feature X.

New examples are generated using the following formulas:

$$V_i^{'} = V_i + N(0, \sigma_V \cdot \alpha), H_i^{'} = H_i + N(0, \sigma_H \cdot \alpha), \quad (2)$$

where:

- $V_i^{'}, H_i^{'}$ — newly generated values of magnetic field anomaly and height for the i-th sample.
- V_i, H_i — values sampled from an existing record in the subgroup.
- $\alpha = 0.1$ — noise intensity coefficient (empirically selected).
- $N(0, \sigma)$ — normally distributed random value with mean 0 and standard deviation σ.

To ensure the physical plausibility of generated values, clipping was applied to constrain them within real-world sensor bounds:

$$V_i^{'} \in [0.0V, 10.6V], H_i^{'} \in [0.0cm, 20.0cm], \tag{3}$$

in accordance with the sensor specifications.

For each subgroup defined by (S, M), 50 new samples were generated, which significantly increased the number of training examples and smoothed the data distribution.

4.2. Data Preprocessing before training

Before training the neural network, the following preprocessing steps were performed:

1. **Normalization** of magnetic anomaly feature *V*:

$$V' = \frac{V - \overline{V}}{\sigma_V}, \tag{4}$$

where:

- V' — normalized magnetic anomaly value.
- V — original value of the magnetic anomaly.
- $\overline{V}$ — mean magnetic anomaly over the entire dataset.
- σ_V — standard deviation of the magnetic anomaly.

2. **Categorical encoding** of the soil type variable S, which takes six values:

- "Dry and Sandy",
- "Dry and Humus",
- "Dry and Limy",
- "Humid and Sandy",
- "Humid and Humus",
- "Humid and Limy".

These categories were encoded using **One-Hot Encoding**, which transforms each category into a binary vector of size six. For example, if the soil type is the second category ("Dry and Humus"), the vector would be:

$$[0,1,0,0,0,0]. \tag{5}$$

3. **Target encoding**. The mine types M, originally in categorical form, were first mapped to numerical indices (0–3) and then encoded using one-hot encoding for input into the neural network.

This method preserved the internal structure and semantics of the data, ensured physical interpretability of the generated values, and significantly improved the model's generalization potential.

5. Mathematical Model of Kolmogorov–Arnold Networks

5.1. Classification Problem Statement

The articles [3, 4], addresses a multiclass-classification task solved with classical machine-learning techniques — artificial neural networks [14] and their variants, support-vector machines [15], Bayesian approaches [16], decision trees [17], and others.

Let X be the feature space X = {V, H, S}, where V denotes the magnetic-field anomaly in the vicinity of a mine (volts), H is the sensor height above the ground that covers the mine, and S represents the soil type. The label set is Y = {0, 1, 2, 3}, whose elements correspond to the classes "no mine," "anti-tank mine," "anti-personnel mine" and "booby-trap" respectively.

The classification objective is to determine a mapping operator Y*: X ⟶ Y that assigns any previously unseen object x ∈ X to class y ∈ Y while minimizing the Euclidean error

$$min \| y^* - y \|, \tag{6}$$

where y is the true class label and y^* is the neural-network prediction.

5.2. KAN Morphology

The Kolmogorov–Arnold Network (KAN) is a neural architecture inspired by the Kolmogorov–Arnold representation theorem [18]. Unlike traditional MLPs that apply fixed nonlinearities at nodes and learn linear weights, KANs apply **learnable nonlinear activation functions on edges**, modeled as univariate splines. Each layer in a KAN consists of **a matrix of spline functions**, and each neuron simply sums the outputs of these spline-parameterized edges.

General Architecture

We define a KAN with shape (3, 16, 16, 4) (Fig. 1), which consists of:

- Input layer with 3 nodes (features).
- Two hidden layers with 16 nodes each.
- Output layer with 4 nodes (classes or regression outputs).

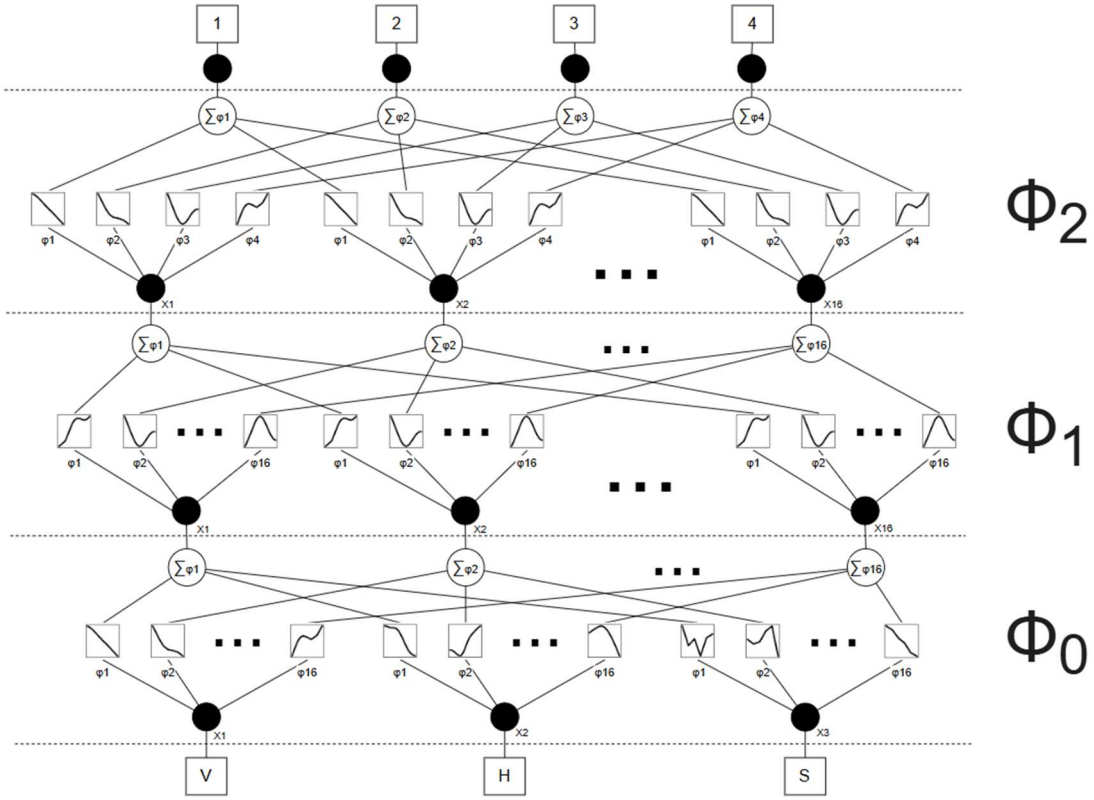

Figure 1: General architecture of KAN.

The general forward propagation is expressed by the composition of KAN layers:

$$KAN(x) = (\Phi_2 \circ \Phi_1 \circ \Phi_0)(x), \tag{7}$$

where each Φ_l is a **functional matrix** consisting of learnable spline activations. Every layer transforms its input by applying these univariate spline functions on each edge, followed by summation at the next layer's nodes.

5.3. Layer-wise Formulation

Let the input vector be $x^{(0)} = x \in \mathrm{R}^3$. The subsequent layer computations are defined as follows:

First Hidden Layer (Layer 0 → 1)

For each neuron $j = 1,\ldots,16$ in the first hidden layer:

$$x_j^{(1)} = \sum_{i=1}^{3} \varphi_{j,i}^{(0)}\left(x_i^{(0)}\right). \tag{8}$$

Second Hidden Layer (Layer 1 → 2)

For each neuron $k = 1,\ldots,16$ in the second hidden layer:

$$x_k^{(2)} = \sum_{j=1}^{16} \varphi_{k,j}^{(1)}\left(x_j^{(1)}\right). \tag{9}$$

Output Layer (Layer 2 → 3)

For each output node $m = 1,\ldots,4$:

$$y_m^{(3)} = \sum_{k=1}^{16} \varphi_{m,k}^{(2)}\left(x_k^{(2)}\right). \tag{10}$$

The final output of the model is:

$$KAN(y) = \begin{bmatrix} y_1^{(3)} \\ y_2^{(3)} \\ y_3^{(3)} \\ y_4^{(3)} \end{bmatrix} \in \mathrm{R} \tag{11}$$

5.4. Spline Activation Functions

Each edge activation function $\varphi_{j,i}^{(l)}(x)$ is defined as a combination of a residual nonlinear term and a cubic B-spline [19]:

$$\varphi(x) = w_b \cdot \mathrm{silu}(x) + w_s \cdot \sum_{m=0}^{G+k-1} c_m B_m(x), \tag{12}$$

where:

- $silu(x) = \dfrac{x}{1+e^{-x}}$ is the smooth SiLU function (acts as a residual base).
- $B_m(x)$ are cubic B-spline basis functions (order $k = 3$).
- $G = 10$ is the number of intervals → $G + k = 13$ basis functions per spline.
- c_m are trainable spline coefficients.
- w_b, w_s are trainable scalar weights controlling the contribution of the SiLU and the spline.

5.5. Parameter Count

To compute the total number of parameters:

- Each spline has $G + k = 13$ coefficients, plus 2 weights (w_b, w_s) → **15 parameters per edge**.
- First layer (φ_0): $3 \times 16 = 48$ edges → $48 \cdot 15 = 720$ parameters

- Second layer $(\varphi_1): 16 \times 16 = 256$ edges $\rightarrow$ $256 \cdot 15 = 3840$ parameters
- Third layer $(\varphi_2): 16 \times 4 = 64$ edges $\rightarrow$ $64 \cdot 15 = 960$ parameters

Total parameters $= 720 + 3840 + 960 = 5520$

6. Computer experiment and results discussion

6.1. Computer experiment

In this section, we present the setup and execution of the computer experiment aimed at evaluating the performance of Kolmogorov–Arnold Networks (KANs) for a multiclass classification task. The experiment involved training and comparing two network architectures: KAN (3, 16, 16, 4) and KAN (3, 64, 64, 4). The architectures were chosen based on the task's dimensionality, where the input space was three-dimensional, and the output space consisted of four distinct classes.

The training was conducted using the PyKAN library [20] on the Python platform. The settings for the networks included the use of cubic B-splines as basis functions, with the order set to 3 and the grid size set to 10. These parameters provided a sufficient balance between the flexibility of the spline approximation and computational efficiency.

The network KAN (3, 16, 16, 4) was trained first. It underwent a training process over a 65 epochs and achieved an accuracy of 93.56% on the test set. In the second case, the KAN (3, 64, 64, 4) architecture was trained. Due to its significantly higher number of parameters, the training took a considerably longer time; however, it achieved an improved test accuracy of approximately 95,59%.

During the training process, loss and accuracy curves were recorded for each model to monitor convergence dynamics and to detect potential signs of overfitting. After the evaluation phase, confusion matrices were generated to provide a detailed understanding of classification performance across all classes. In addition to the visual analyses, a comprehensive classification report was produced, presenting key metrics such as precision, recall, and F1-score for each class.

The details of the experimental environment, including the software tools and libraries, are as follows:

Programming language: Python

Neural network library: PyKAN [20]

Hardware: Personal PC (AMD Ryzen 5 5600G CPU, NVIDIA GeForce RTX 4060 GPU)

Software:

- IntelliJ IDEA (with Python plugin support).
- Python 3.x.
- PyKAN library [20].
- CUDA Toolkit (for GPU acceleration with NVIDIA RTX 4060).
- PyTorch (backend library for PyKAN).
- NumPy (for data manipulation).
- Matplotlib (for visualization of results).

6.2. Discussion of Results

The results of the experiments are illustrated through loss-accuracy curves, confusion matrices, and a set of other performance metrics [21], which comprehensively describe the behavior of both tested architectures.

For the KAN (3, 16, 16, 4) network, the loss curve (Figure 2) demonstrated a steady decrease without abrupt oscillations, indicating stable convergence. The corresponding accuracy curve (Figure 2) showed consistent improvement throughout the training process, reaching a plateau near 92.56%. The confusion matrix (Figure 3 and Figure 4) revealed that most misclassifications occurred between the (specify which classes if possible), suggesting that the network found these classes harder to differentiate given the feature space.

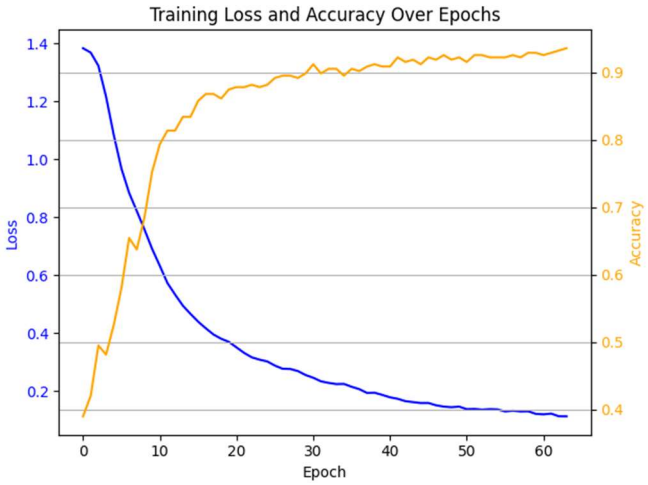

Figure 2: Training accuracy and loss over epochs KAN (3, 16, 16, 4).

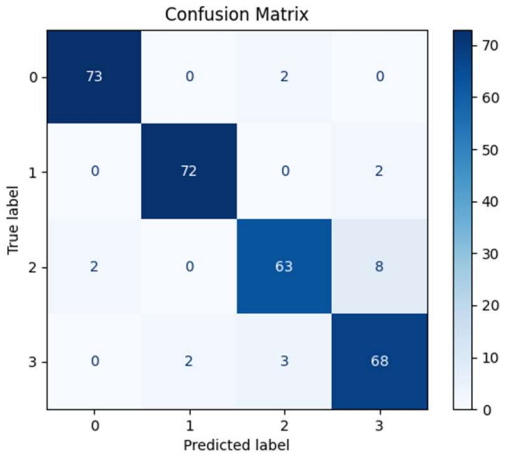

Figure 3: Confusion matrix KAN (3, 16, 16, 4).

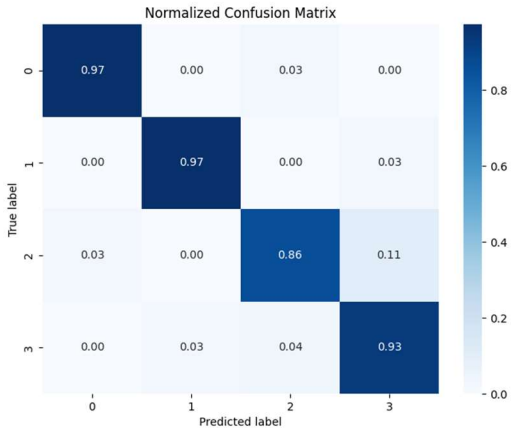

Figure 4: Normalized confusion matrix KAN (3, 16, 16, 4).

The normalized confusion matrix for the KAN (3, 16, 16, 4) model, shown in Figs. 3 and 4, reflects almost perfect identification of the "no mine" and "anti-tank" classes, with correct detection rates reaching approximately ninety-seven percent. At the same time, the majority of misclassifications occurred between the "anti-personnel" and "booby trap" classes: around ten

objects from the first category were confused with the second, while the reverse misclassification happened almost twice as rarely. This asymmetry is explained by the partial overlap of magnetic anomaly ranges and sensor height, indicating that the three-dimensional feature space was insufficient to fully separate these mine types.

Despite this, the model exhibits stable convergence of the loss function and absence of sharp fluctuations, indicating proper hyperparameter tuning and sufficient capacity for the basic task. However, it also signals the need to enrich the feature space specifically in the area where classification errors are observed.

Table 4
Classification Report KAN (3, 16, 16, 4)

Class	Precision	Recall	F1-score	Support
0	0.9733	0.9733	0.9733	75
1	0.9730	0.9730	0.9730	74
2	0.9265	0.8630	0.8936	73
3	0.8718	0.9315	0.9007	73
Accuracy	-	-	0.9356	295
Macro avg	0.9361	0.9352	0.9351	295
Weighted avg	0.9365	0.9356	0.9355	295

Columns:

- **Precision** - The proportion of predicted positive samples that are actually correct for each class.
- **Recall** - The proportion of actual positive samples that are correctly predicted for each class.
- **F1-score** - The harmonic mean of precision and recall for each class, providing a balance between the two metrics.
- **Support** - The number of true instances for each class in the test set.
- **Precision -** The proportion of predicted positive samples that are actually correct for each class.

Rows:

- **0, 1, 2, 3** - The performance metrics for each individual class.
- **Accuracy** - The overall classification accuracy across all classes (i.e., the proportion of correctly classified samples).
- **Macro avg** - The unweighted mean of precision, recall, and F1-score across all classes, treating each class equally regardless of its support.
- **Weighted avg** - The mean of precision, recall, and F1-score weighted by the number of true instances (support) for each class, giving more influence to classes with more samples.

On the other hand, the KAN (3, 64, 64, 4) architecture, while requiring longer training time due to the increased number of neurons, achieved superior classification results with approximately 95% accuracy. Its loss curve (Figure 5) exhibited a smoother descent, and its accuracy curve (Figure 5) achieved a slightly higher and more stable plateau compared to the smaller network. The confusion matrix (Figure 6 and Figure 7) for this model showed a significant reduction in misclassification rates across all classes, particularly improving recognition of (specify if needed).

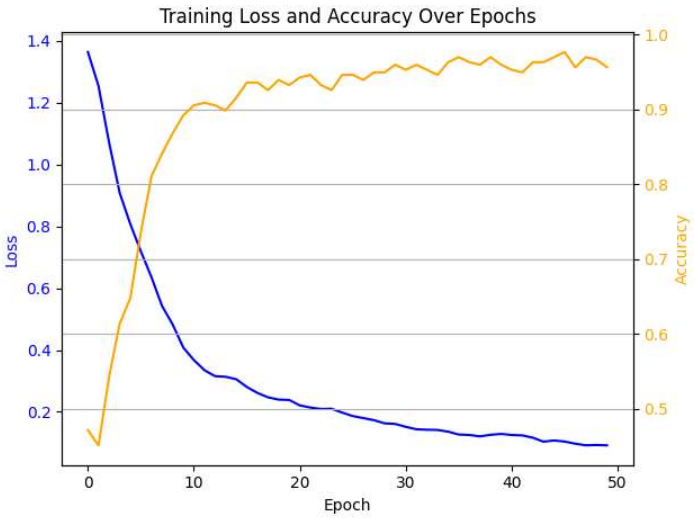

Figure 5: Loss and accuracy over epochs KAN (3, 64, 64, 4).

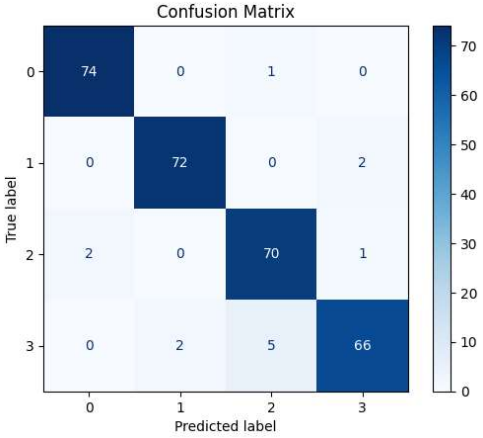

Figure 6: Confusion matrix KAN (3, 64, 64, 4).

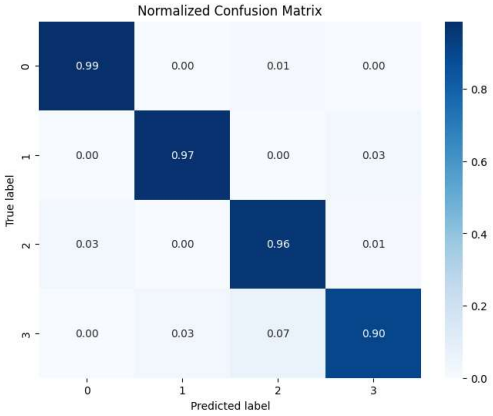

Figure 7: Normalized confusion matrix KAN (3, 64, 64, 4).

Increasing the number of neurons to sixty-four in each hidden layer led to significant changes in the error patterns, as clearly seen in the confusion matrices in Figs. 6 and 7. The updated KAN (3, 64, 64, 4) architecture almost completely eliminated confusion between the "anti-personnel" and "booby trap" classes in the direction from the latter to the former, raising the accuracy for the "booby trap" class above ninety-five percent. Reverse confusion still occurred in about seven cases

out of seventy-three, reducing the recall of this class to ninety-four percent, but these mistakes now have a one-sided nature: the network becomes more conservative, assigning doubtful samples to the less dangerous category in the absence of convincing evidence. The increased computational costs are justified by the fact that overall classification accuracy improved by about two percent, and the off-diagonal elements of the matrix sharply decreased for all classes except for the localized issue of booby trap identification.

The comparison of the two models shows that even with the same basic set of features, a wider architecture can capture finer signal nonlinearities and thus reduce the number of critical errors. At the same time, the remaining confusion between classes 2 and 3 indicates the limit beyond which pure network scaling becomes less effective compared to introducing additional information, such as gradient characteristics of the magnetic field or contextual soil indicators.

Thus, detailed analysis of the confusion matrices indicates that the main direction for further optimization should be strengthening the discriminative power of features specifically for the "booby trap" class, while preserving the already achieved high reliability in detecting other mines and safe areas.

Table 5
Classification report KAN (3, 64, 64, 4)

Class	Precision	Recall	F1-score	Support
0	0.9737	0.9867	0.9801	75
1	0.9730	0.9730	0.9730	74
2	0.9211	0.9589	0.9396	73
3	0.9565	0.9041	0.9296	73
Accuracy	–	–	0.9559	295
Macro avg	0.9561	0.9557	0.9556	295
Weighted avg	0.9562	0.9559	0.9558	295

Comparative analysis of the two models (Table 4 and Table 5) indicates that increasing the hidden layer size improves generalization capability but at the cost of greater computational time and resources. This trade-off must be considered depending on the application domain requirements.

Thus, the computer experiments have demonstrated that Kolmogorov–Arnold Networks, when properly configured with cubic B-splines and an appropriate grid resolution, can achieve high accuracy in multiclass classification problems, with performance scaling positively with network capacity.

Study Limitations

The base dataset has a limited volume; although synthetic augmentation improves generalization, it cannot fully replace field measurements. The results were obtained under laboratory conditions without considering the influence of metallic debris, heterogeneous magnetic backgrounds, or sensor temperature drifts.

Future Research Directions

Collection of large-scale field data under various climatic and geological conditions to validate the results.

End-to-end optimization: selection of spline grids, nonlinearity bases, and regularization techniques (e.g., KAN-Mixer, weight priorities) to further improve accuracy without exponential growth in parameters.

Robustness: investigation of resilience to adversarial influences typical of deceptive mine masking with metallic shrapnel or geomagnetic traps.

7. Conclusions

1. For the first time, Kolmogorov–Arnold Networks (KAN) with cubic B-splines were used for passive mine recognition based on magnetic anomalies. Unlike classical MLPs, KAN allows modeling nonlinear dependencies at the level of weight connections, enhancing the interpretability and robustness of the model to noise in real sensor measurements.

2. An extended dataset was created: synthetic samples were added to 338 original magnetic anomaly recordings, generated using a parameterized normal distribution with a step of 50 samples for each "soil type – mine type" subset. This balanced the feature variance and reduced the risk of overfitting.

3. Two architectures were developed: KAN (3, 16, 16, 4) and KAN (3, 64, 64, 4). Both models were trained using the PyTorch-compatible PyKAN library with identical spline hyperparameters.

KAN (3, 16, 16, 4) achieved 93.56% accuracy without signs of overfitting; the main errors occurred between the "anti-personnel" and "booby trap" classes. Increasing the number of neurons in KAN (3, 64, 64, 4) to 64 per hidden layer improved accuracy to 95.59%, significantly reducing false detections across all four classes. The cost of this improvement was an almost linear increase in the number of parameters and training time. This confirms the advisability of adaptively selecting model size based on the hardware constraints of field systems. Confusion-matrix analysis showed both models nearly flawless at identifying “no-mine” and anti-tank cases, while most errors arose from confusion between anti-personnel mines and booby-traps. Results confirm that increasing network capacity improves discrimination among visually similar magnetic signatures.

4. Advantages of the proposed approach.

Passive mine recognition: the use of the FLC-100 sensor does not require active excitation, minimizing the risk of detonation.

Interpretability: spline weights enable analysis of the contribution of each feature and facilitate safety certification.

Robustness: experiments showed no sharp fluctuations in the loss function and stable convergence even on a noise-enriched dataset.

The study proves that properly configured Kolmogorov–Arnold Networks can achieve over 95% accuracy in multiclass mine classification based on passive magnetic features. The combination of interpretable spline weights, high sensitivity to small anomalies, and scalability potential makes KAN a promising foundation for modular humanitarian demining systems, which can significantly accelerate land clearance and reduce risks for personnel.

Acknowledgements

The research was carried out with the grant support of the National Research Fund of Ukraine "Methods and means of active and passive recognition of mines based on deep neural networks", project registration number 273/0024 from 1/08/2024 (2023.04/0024). Also, we would like to thank the reviewers for their precise and concise recommendations that improved the presentation of the results obtained.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] Landmine and Cluster Munition Monitor, *Landmine Monitor 2023*, International Campaign to Ban Landmines – Cluster Munition Coalition (ICBL-CMC) (2023). URL: https://backend.icblcmc.org/assets/reports/Landmine-Monitors/LMM2023/Downloads/Landmine-Monitor-2023_web.pdf

[2] V. Lytvyn, I. Peleshchak, R. Peleshchak, O. Mediakov, P. Pukach, *Development of a hybrid neural network model for mine detection by using ultrawideband radar data*, Eastern-European Journal of Enterprise Technologies 3 (2023) 78–85. doi:10.15587/1729-4061.2023.279891.

[3] R. M. Peleshchak, V. V. Lytvyn, M. A. Nazarkevych, I. R. Peleshchak, H. Y. Nazarkevych, *Influence of the Symmetry Neural Network Morphology on the Mine Detection Metric*, Symmetry 16 (2024) 485. doi:10.3390/sym16040485.

[4] C. Yilmaz, H. T. Kahraman; S. Söyler, *Passive Mine Detection and Classification Method Based on Hybrid Model*, IEEE (2018). doi:10.1109/ACCESS.2018.2866538.

[5] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, M. Tegmark, *KAN: Kolmogorov–Arnold Networks*, arXiv (2024). doi:10.48550/arXiv.2404.19756.

[6] J. Erdmann, F. Mausolf, J.L. Späh, *KAN we improve on HEP classification tasks? Kolmogorov-Arnold Networks applied to an LHC physics example*, arXiv (2024). doi: 10.48550/arXiv.2408.02743.

[7] S. Somvanshi, S. A. Javed, M. M. Islam, D. Pandit, S. Das, *A Survey on Kolmogorov-Arnold Network*, arXiv (2023). doi:10.48550/arXiv.2411.06078.

[8] I. Barašin, B. Bertalanič, M. Mohorčič, C. Fortuna, *Exploring Kolmogorov-Arnold Networks for Interpretable Time Series Classification*, arXiv (2024). doi:10.48550/arXiv.2411.14904.

[9] S. Wang, X. Yue, Y. Lyu, C. Shan, *Exploring Adversarial Transferability between Kolmogorov-arnold Networks*, arXiv (2025). doi:10.48550/arXiv.2503.06276.

[10] M. Cheon, *Kolmogorov-Arnold Network for Satellite Image Classification in Remote Sensing*, arXiv (2024). doi:10.48550/arXiv.2406.00600.

[11] I. Drokin, *Kolmogorov-Arnold Convolutions: Design Principles and Empirical Studies*, arXiv (2024). doi:10.48550/arXiv.2407.01092.

[12] GmbH & Co. KG, *Magnetic Field Sensor FLC 100*, n.d. URL: https://www.scribd.com/document/382854526/Data-Sheet-FLC-100.

[13] K. Maharana, S. Mondal, B. Nemade, *A review: Data pre-processing and data augmentation techniques*, Global Transitions Proceedings 3 (2022) 91–99. doi:10.1016/j.gltp.2022.04.020.

[14] R. Qamar, B. A. Zardari, *Artificial Neural Networks: An Overview* (2023). doi:10.58496/mjcsc/2023/015.

[15] L. Vanneschi, S. Silva, *Support Vector Machines*, Springer (2023). doi:10.1007/978-3-031-17922-8_10.

[16] J. Lampinen, A. Vehtari, *Bayesian approach for neural networks — review and case studies*, Neural Networks (2001). doi:10.1016/S0893-6080(00)00098-8.

[17] B. de Ville, *Decision trees*, Wiley Interdiscip. Rev. Comput. Stat. (2013). doi:10.1002/wics.1278.

[18] J. Schmidt-Hieber, *The Kolmogorov-Arnold representation theorem revisited*, arXiv (2021). doi:10.48550/arXiv.2007.15884.

[19] A. Chaudhuri, *B-Splines*, arXiv (2021). doi:10.48550/arXiv.2108.06617.

[20] KindXiaoming, *PyKAN: Kolmogorov–Arnold Networks in PyTorch*, GitHub Repository (2024). URL: https://github.com/KindXiaoming/pykan.

[21] R. C. Eberhart, R. W. Dobbins, L. V. Hutton, *Performance Metrics*, in: R. C. Eberhart, R. W. Dobbins (Eds.), Neural Network PC Tools, Academic Press, 1990, 161–176. doi:10.1016/B978-0-12-228640-7.50013-1.