Radar Target Recognition Based on Machine Learning

Volodymyr Motyka[1], Mariia Nasalska[1], Yaroslav Stepaniak[1], Victoria Vysotska[1,2] and Myroslava Bublyk[1]

[1] *Lviv Polytechnic National University, S. Bandera Street, 12, Lviv, 79013, Ukraine*
[2] *Osnabrück University, Friedrich-Janssen-Str. 1, Osnabrück, 49076, Germany*

Abstract

The intelligence of the air situation is based on radar information about the air enemy, which allows you to reveal the raid's target, determine the composition and means that take part in the raid, determine the most dangerous means and ensure that weapons are aimed at them. The article's main goal is to build a target recognition model in the form of AGM-86C (CALCM) and CR Taurus KEPD 350 cruise missiles. A module for recognising AGM-86C and Taurus KEPD 350 cruise missiles have also been built, which sufficiently accurately recognises these types of missiles (accuracy - 84.52%). Because one of the missiles is sometimes referred to as a missile developed using "stealth" technology, the model can be considered effective. The disadvantage of the model is that the model is trained on only two types of missiles. The problem is that the data on the effective scattering surface of all missiles in any state's arsenal is top-secret data.

Keywords 1

Target recognition, radar, missile, machine learning, decision tree, classification task, model training

1. Introduction

February 24, 2022, changed the lives of people in Ukraine to "before" and "after". The war is being waged on all fronts: front lines, rear, and abroad. At such a time, people do everything that depends on them to help the state, and more and more people come from abroad to Ukraine, despite the constant danger from the invaders' side. Almost every day, air alarms sound, and various missiles are aimed at our rear, including civilian objects. Air defences work extremely well. The problem is that each missile is different in its properties, and this causes certain problems when intercepting them. That is why the preliminary recognition of air targets is an extremely important aspect for the protection of the rear and, in particular, the front.

2. Related works

Automatic target recognition is the ability of an algorithm or device to recognise targets or objects based on data received from sensors. It is used primarily for military and military purposes. This is an extremely wide range of algorithms and devices, most of which data is classified [1].

The simplest version of automatic target recognition is a radar recognition system, a hardware and software technical complex for automatically distinguishing one's troops and weapons from the enemy [2].

IntelITSIS'2023: 4th International Workshop on Intelligent Information Technologies and Systems of Information Security, March 22–24, 2023, Khmelnytskyi, Ukraine
EMAIL: volodymyr.motyka@lpnu.ua (V. Motyka), mariia.nasalska@lpnu.ua (M. Nasalska), yaroslav.stepaniak@lpnu.ua (Y. Stepaniak), victoria.a.vysotska@lpnu.ua (V. Vysotska), my.bublyk@gmail.com (M. Bublyk)
ORCID: 0000-0002-7826-030X (V. Motyka), 0000-0001-9969-1937 (M. Nasalska), 0000-0001-6166-3500 (Y. Stepaniak), 0000-0001-6417-3689 (V. Vysotska), 0000-0003-2403-0784 (M. Bublyk)

The essence of recognition is to establish whether a target belongs to a certain class based on the results of processing radar signals reflected or emitted from it [3]. In this way, the "Home-Alien" systems are divided into active (the target emits the signal) and passive (the signal is reflected from the target). Let's define the concept of "Radiolocation".

Radiolocation is detecting objects (targets) and determining their spatial coordinates and movement parameters using technical radio means and methods. This process is called radar surveillance, and the devices for this purpose are radar stations (RADS) or radars [4].

The main characteristics of air attack vehicles (AAV) as radar objects are the following [5]:

- Effective scattering surface of the target (ESS);
- Diagram of reverse secondary radiation;
- Long-range radar portrait of the target.

The effective scattering surface (ESS) of the target is the main energy characteristic of the target. It affects the target detection range of the radar station and the quality of radar information [6].

The effective scattering surface is the area of such an equivalent secondary emitter, which uniformly scatters all the electromagnetic energy falling on it, and creates the same energy power flow density at the reception point as the real target [7-8].

The effective scattering area (ESA) is a characteristic of the target reflectivity, which is determined by the power ratio of the electromagnetic energy reflected by the target in the direction of the receiver (radar) to the surface density of the energy flow of the incident plane wave [9].

EPR of the target depends on the following factors [1-9]:

- Physical geometry and external features of the target;
- Radar direction;
- Frequency of radar transmitters;
- Electrical properties of the target surface.

While the design of passenger aircraft is more focused on efficiency and safety, in the case of aircraft used for military purposes, care is taken to keep this reflective surface as small as possible. Measures to achieve this are called stealth technology.

The ESA of the target is the size of the metal surface from which the secondary radiation flux creates the same flux at the reception point as from the real target.

The magnitude of ESA depends on the electrical properties of the material of the aircraft structure and its coating, the ratio of its geometric dimensions and wavelength, the angle of irradiation, and the polarisation of electromagnetic waves irradiating the target. Moreover, it is inherent in the complex objective to change the polarisation of the incident wave, due to which components orthogonal to the probing signal appear in the reflected signal.

The ESA will be different for different classes of aircraft, ranging from 1 to 10м^2 for tactical aircraft and more than 10м^2 for bombers. For aircraft developed using "Stealth" technology, the ESA is the value 0,1…1,0м^2 [6-9].

ESA can be calculated using the following formula [6]:

$$\sigma_t = 4\pi R^2 \cdot \frac{S_r}{S_t}, \quad (1)$$

where S_r is radio-wave energy flow density near the radar receiver; S_t is radio-wave energy flow density near the target; R is the distance from the target to the radar receiver.

A smooth, flat surface that conducts perfectly has a narrow secondary scattering pattern. The main energy part of the reflected wave is contained in the main petal of the antenna direction diagram (DAD). The ADD width decreases according to the size increase of the reflecting surface and the length shortening of the incident wave. If the surface is irradiated at a right angle, the reflected energy's main part returns to the radiation source. At irradiation angles of less than 90°, the scattered energy part within the side lobes of the ADD is returned to the radar [10-16].

Complex objects (planes, ships, tanks) can be considered a collection of many separate elements that disperse electromagnetic energy in different directions. The total amplitude of the reflected signal is determined by the relative phases and amplitudes of the emissions of individual reflectors and is subject to fluctuations. The nature of the fluctuation of the resulting signal largely depends on the object's speed and direction of movement and even its individual elements relative to the radar [17-19]. The phases of signals reflected by complex targets also change. In electromagnetic waves (EMW) scattering by

various objects, depolarisation of signals usually occurs [20-24]. Scattering diagrams of real objects show the scattering intensity's dependence on the wave's angle of incidence, determined by their configuration and orientation relative to the radar station. As a rule, they are multi-petaled.

In modern AAV, the ESA varies widely, and the properties of the target type, in turn, significantly depend on the wavelength of the radar, the angle of target irradiation and some other factors [25-29]. ESA drops can reach 100...1000 or more times [30-33]. The latest aircraft and UAVs (unmanned aerial vehicles) are developed considering the reduction of their ESA. Thus, the geometric dimensions of the B-1B aircraft are 10% smaller than the geometric dimensions of the B-52 aircraft, and its ESA is almost 10 times smaller. The ESA of fighters, cruise and guided missiles also decreased significantly.

The main task is to build several models and choose the best of the built ones. The essence of the models is the recognition of cruise missiles (CR) AGM-86C(CALCM) [9] and CR Taurus KEPD 350 [12-14]. The calculation of the effective scattering surface (ESS) is carried out at two frequencies: f= 180 MHz ($\lambda_1 \approx 1.667$ m), which corresponds to the operating frequency range of the 5N84A, P-18 radar standby mode, and f= 350 MHz ($\lambda_1 \approx 0.857$ m) for comparison.

The azimuth was calculated in the horizontal plane (wing plane) from the nose angle (= 0 corresponds to the narrowing "in the nose" of the radar, = 180 degrees. - sounding "in the tail"). The azimuth change step was 0.5 degrees. For each fixed value, the sounding location angle β was chosen randomly, uniformly distributed in the range ε=-3±4 degrees relative to the wing plane (a negative location angle corresponds to the probing part of the lower hemisphere). The results are obtained for two polarisations - horizontal (the electric field strength vector of the sounding signal is parallel to the wing plane) and vertical (the electric field strength vector of the sounding wave lies in the plane, orthogonal to the wing plane and passing through the sounding direction vector).

3. Methods

We have two types of missiles and calculated effective scattering areas at different elevation angles (RADS) and azimuths. In other words, we solve the problem of classification. For this, we will use the decision tree method.

A decision tree represents a sequence of decisions and environmental states, indicating the corresponding probabilities and payoffs for any combination of alternatives and environmental states.

On the edges ("branches") of the decision-making tree, the attributes on which the objective function depends are written; in the "leaf", the values of the objective function are written, and in other nodes - the attributes by which the cases are distinguished. To classify a new case, it is necessary to go down the tree to the letter and issue the appropriate value.

Each leaf represents the value of the target variable, changed during the movement from the root to the leaf. Each internal node corresponds to one of the input variables. A tree can also be "learned" by dividing the original variables into subsets based on testing attribute values. This process is repeated on each of the resulting subsets. The recursion terminates when the subset at a node has the same values as the target variable, so it adds no value to predictions.

A rule is a logical construction presented as "if: then:".

Decision trees have several advantages when used.

- The user easily interprets trees, and they are intuitive.
- Decision trees enable extracting rules from a database in plain language. Example rule: If Age >35 and Income >200, issue a loan.
- Decision trees make it possible to create classification models in those areas where it is quite difficult for the analyst to formalise knowledge.
- The decision tree construction algorithm does not require the user to select input attributes (independent variables). All existing attributes can be submitted to the input of the algorithm, the algorithm itself will select the most significant among them, and only they will be used to build a tree. Compared, for example, with neural networks, this makes the user's work much easier since the choice of the number of input attributes significantly affects the training time in neural networks.
- Models created using decision trees have higher accuracy than other methods of building classification models (statistical methods, neural networks).

- Fast learning process. Building classification models using decision tree algorithms takes significantly less time than, for example, training neural networks.
- Most algorithms for constructing decision trees can especially handle missing values.
- Many classical statistical methods used to solve classification problems can only work with numerical data, while decision trees work with both numerical and categorical data types.

4. Experiments, results and discussions

Data for processing is not publicly available (four files in .xls format as agm86c_f_180_vertical, agm86c_f_180_horizontal, kepd350_f_180_vertical, kepd350_f_180_horizontal). Each file contains four columns of data:

89.01014736	0	32.34668541	0.000285704
90.54658416	0.5	32.39473194	0.000930973
93.68004879	1	31.33735002	0.009354365
91.80246483	1.5	31.38034445	0.012273822
95.5827018	2	28.93900567	0.085906038
90.39303196	2.5	29.78845157	0.041796866
94.68396326	3	27.435552	0.144342568
91.43188842	3.5	27.26550993	0.073528013
89.73129015	4	25.90301595	0.132250099
90.1785073	4.5	24.3586741	0.143169677
96.90806779	5	20.27232615	0.756644658
89.95263786	5.5	21.0039888	0.221759192
89.07138132	6	19.23492231	0.354000132
93.25331327	6.5	16.98907075	0.377292482
93.81411307	7	15.1136725	0.525210862
90.32999586	7.5	14.05407717	0.336499036
92.60628714	8	12.16667148	0.431644639
89.45644718	8.5	10.93426729	0.525907196
95.26655336	9	8.773305575	1.279912017
93.15900926	9.5	8.032427675	0.673798421
96.00774577	10	6.535233374	1.834122403
96.64719865	10.5	5.669434237	2.300434511
93.31473202	11	5.500640596	0.862906533
92.69659065	11.5	5.071053012	0.73677336
95.89775625	12	4.755947889	2.177824883
95.23726651	12.5	4.833353202	1.869880218
96.97436515	13	5.056289418	3.034274924
93.89194151	13.5	5.444373565	1.289186448
91.12971125	14	5.646353262	0.619165174
95.72095306	14.5	6.993791059	2.353914433
92.00685855	15	7.427189592	0.743224061

Figure 1: The view of the soars data (Screenshot of part of the input data)

The first column describes the elevation angle, the second - the azimuth, and the third and fourth - the calculated ESA of the missiles. In the third column, named ESA, data is at a frequency of f = 180 MHz, and in the fourth, for comparison, ESA at f=350 MHz. To use the decision tree, we will combine the data into one table, where the columns will describe a certain missile at a certain frequency and polarisation.

- Angle - the angle of the place, azimuth – azimuth, RCS (radar cross section),
- H RCS (f = 180 MHz) – EPR at a frequency of 180 MHz and horizontal polarisation,
- H RCS (f = 350 MHz) – EPR at a frequency of 350 MHz and horizontal polarisation,
- V RCS (f = 180 MHz) – EPR at a frequency of 180 MHz and vertical polarisation,
- V RCS (f = 180 MHz) – ESR at a frequency of 350 MHz and vertical polarisation,
- Missile is a type of missile (that takes two values)

As a result, we have 722 lines of data (361 lines for each missile).

Let's reformat the file into a .csv file and start the main part of the work - building models.

Angle	Azimuth	H RCS (f = 180 MHz)	H RCS (f = 350 MHz)	V RCS (f=180 MHz)	V RCS (f=350 MHz)	missile
89.01015	0	2.0513996068	0.0000000000	3.8041211678	0.0000000000	agm86c
90.54658	0.5	1.7346564667	0.0022137812	3.7449501630	0.0004226931	agm86c
93.68005	1	1.2725430814	0.0126922654	3.5394311553	0.0052750432	agm86c
91.80246	1.5	1.5444753525	0.0226887686	3.5477221929	0.0064012921	agm86c
95.5827	2	0.9758979636	0.0633339884	3.2293332472	0.0334369731	agm86c
90.39303	2.5	1.7709292154	0.0529584752	3.3291808091	0.0093988707	agm86c
94.68396	3	1.1281533326	0.1259716703	3.0313029436	0.0604491158	agm86c
91.43189	3.5	1.6110615821	0.1134023927	2.9979670750	0.0286350590	agm86c
89.73129	4	1.9240657625	0.1211462218	2.8720531553	0.0153541222	agm86c
90.17851	4.5	1.8408639576	0.1572084332	2.7465667575	0.0243573878	agm86c
96.90807	5	0.8104282904	0.4387055010	2.6249406490	0.2588667235	agm86c
89.95264	5.5	1.9156094570	0.2191340468	2.6200862614	0.0290946892	agm86c
89.07138	6	2.1647064585	0.2339219521	2.6224840473	0.0168133994	agm86c
93.25331	6.5	1.3964945007	0.4405417633	2.8288695480	0.1714481325	agm86c
93.81411	7	1.3305444955	0.5396612311	3.0551139985	0.2314306077	agm86c
90.33	7.5	1.9163465266	0.3791441724	3.2072381449	0.0552446504	agm86c
92.60629	8	1.5390849386	0.5645513162	3.6986627177	0.1898440418	agm86c
89.45645	8.5	2.2136088200	0.4166967027	3.9889011113	0.0290567020	agm86c
95.26655	9	1.2079165246	0.9927375028	4.9336271192	0.5125830668	agm86c
93.15901	9.5	1.5336835076	0.7884585288	5.5053577347	0.2952563276	agm86c
96.00775	10	1.1685148889	1.2890637765	6.4575546177	0.7094478061	agm86c
96.6472	10.5	1.1208382760	1.5081391602	7.3493341676	0.8672493086	agm86c
93.31473	11	1.6190381794	0.9787982833	8.1328306424	0.3680971281	agm86c
92.69659	11.5	1.7591893228	0.9359866317	9.1097198526	0.2970943033	agm86c
95.89776	12	1.3621607680	1.6171995532	10.3262022858	0.8630290831	agm86c
95.23727	12.5	1.5034685137	1.5345522665	11.4155644434	0.7585015912	agm86c
96.97437	13	1.3473103557	2.0609304792	12.4613451656	1.1711653210	agm86c
93.89194	13.5	1.8098640116	1.3368845759	13.6644331932	0.5186073032	agm86c

Figure 2: Processed data (Screenshot of part of the input data)

First, we import the necessary libraries and the dataset itself:

```
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
import seaborn as sns
import matplotlib.pyplot as plt
#import pydotplus
%matplotlib inline
import matplotlib.pyplot as plt
```

```
df.info()
✓ 0.4s
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 722 entries, 0 to 721
Data columns (total 7 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Angle               722 non-null    float64
 1   Azimuth             722 non-null    float64
 2   H RCS (f = 180 MHz) 722 non-null    float64
 3   H RCS (f = 350 MHz) 722 non-null    float64
 4   V RCS (f=180 MHz)   722 non-null    float64
 5   V RCS (f=350 MHz)   722 non-null    float64
 6   missile             722 non-null    object
dtypes: float64(6), object(1)
memory usage: 39.6+ KB
```

	Angle	Azimuth	H RCS (f = 180 MHz)	H RCS (f = 350 MHz)	V RCS (f=180 MHz)	V RCS (f=350 MHz)	missile
0	89.010147	0.0	2.051400	2.660000e-25	3.804121	9.190000e-25	agm86c
1	90.546584	0.5	1.734656	2.213781e-03	3.744950	4.226930e-04	agm86c
2	93.680049	1.0	1.272543	1.269226e-02	3.539431	5.275043e-03	agm86c
3	91.802465	1.5	1.544475	2.268877e-02	3.547722	6.401292e-03	agm86c
4	95.582702	2.0	0.975898	6.333399e-02	3.229333	3.343697e-02	agm86c

Figure 3: Example data and data types of variables (Screenshot of the program)

After analysing the dataset, we will briefly describe the attributes we need for further implementation. We have a total of 722 data records. What is important is that we have only two unique

values of 'missile', namely AMG86c and KEPD350, two independent variables, namely 'Angle' and 'Azimuth', four dependent ones - 'H RCS (f = 180 MHz)', 'H RCS (f = 350 MHz)', 'V RCS (f=180 MHz)', 'V RCS (f=350 MHz)'. How exactly the values of the variables vary can be seen in Figure 5.

	Angle	Azimuth	H RCS (f = 180 MHz)	H RCS (f = 350 MHz)	V RCS (f=180 MHz)	V RCS (f=350 MHz)
count	722.000000	722.000000	722.000000	7.220000e+02	722.000000	7.220000e+02
mean	92.863075	90.000000	5.496159	8.003638e-01	9.603326	4.648339e-01
std	2.280204	52.141784	8.034358	8.538729e-01	10.231695	5.734494e-01
min	89.010147	0.000000	0.021788	2.660000e-25	0.061390	9.190000e-25
25%	90.684599	45.000000	1.281541	1.937950e-01	2.086965	8.622363e-02
50%	93.136960	90.000000	3.078279	4.010581e-01	7.507735	2.320925e-01
75%	94.804597	135.000000	5.391920	1.157273e+00	13.826845	6.429147e-01
max	96.974365	180.000000	44.276757	4.837641e+00	75.750707	3.506987e+00

Figure 4: Description of the dataset (Screenshot of the program)

The next step is to create data groups for training and testing the model. The test sample will consist of 33% of the dataset:

```
X = df.drop(['missile'], axis=1)
y = df['missile']
X_train , X_test, y_train, y_test = train_test_split(X, y, test_size=0.33,
random_state=42)
```

```
X_train.shape, X_test.shape
```

✓ 0.3s

```
((483, 6), (239, 6))
```

```
X_train.head()
```

✓ 0.3s Python

	Angle	Azimuth	H RCS (f = 180 MHz)	H RCS (f = 350 MHz)	V RCS (f=180 MHz)	V RCS (f=350 MHz)
42	93.579190	21.0	3.555251	1.564884	20.340526	0.125035
73	96.863598	36.5	0.649556	2.337733	16.967664	2.320665
167	93.422450	83.5	0.356577	0.086764	3.756759	0.014718
250	89.308655	125.0	0.303236	0.077204	6.884478	0.214986
631	92.530441	135.0	4.263225	1.605771	1.122514	0.057594

```
X_test.head()
```

✓ 0.4s Python

	Angle	Azimuth	H RCS (f = 180 MHz)	H RCS (f = 350 MHz)	V RCS (f=180 MHz)	V RCS (f=350 MHz)
336	95.168060	168.0	6.037746	2.336418	17.540357	0.093433
516	96.359754	77.5	4.834882	0.901656	9.310164	0.984244
552	93.384332	95.5	21.390854	0.277159	1.563715	0.178914
33	91.207098	16.5	2.929826	1.078598	19.699535	0.074136
558	91.889125	98.5	6.404130	0.166498	1.212775	0.212532

Figure 5: Distribution of data and examples of created sample data (Screenshot of the program)

Now let's move on to creating an instance of the DecisionTreeClassifier model with the Gini index criterion and to training this model:

```
clf_gini = DecisionTreeClassifier(criterion='gini', max_depth=6, random_state=0)
clf_gini.fit(X_train,y_train)
y_pred_gini = clf_gini.predict(X_test)
```

We will immediately check the accuracy of the built model (Figure 6):

```
from sklearn.metrics import accuracy_score
print('Model accuracy score with criterion gini index: {0:0:4f}',
format(accuracy_score(y_test, y_pred_gini)))
y_pred_train_gini = clf_gini.predict(X_train)
y_pred_train_gini
print('Training-set accuracy score: {0:0.4f}', format(accuracy_score(y_train,
y_pred_train_gini)))
```

```
Model accuracy score with criterion gini index: {0:0:4f} 0.8451882845188284
Training-set accuracy score: {0:0.4f} 0.9130434782608695
```

Figure 6: Accuracy of the model (Gini criterion)

At this stage, we will build a graph of the decision tree using the following block of code:

```
from sklearn import tree
plt.figure(figsize=(12,8))
tree.plot_tree(clf_gini.fit(X_train, y_train))
plt.show
```

The graph view is not readable, so we export the graph tree solution using the function of the graphviz module. This module allows you to view detailed examples of tree solutions. In Fig. 8, you can see that it deciphers the elements of the model more understandably:

```
features = ['Angle','Azimuth','H RCS (f = 180 MHz)','H RCS (f = 350 MHz)',
'V RCS (f=180 MHz)','V RCS (f=350 MHz)']
classnames = ['agm86c','kepd350']

tree.export_graphviz(clf_gini,
                     out_file="treegini.dot",
                     feature_names = features,
                     class_names=classnames,
                     filled = True)
```

Let's move on to creating an instance of the DecisionTreeClassifier model with the Entropy index criterion and to training this model:

```
clf_en = DecisionTreeClassifier(criterion='entropy', max_depth=6,
random_state=0)
clf_en.fit(X_train, y_train)
y_pred_en = clf_en.predict(X_test)
```

And we will also check the accuracy of the built model (Figure 7):

```
print('Model accuracy score with criterion entropy: {0:0.4f}'.
format(accuracy_score(y_test, y_pred_en)))
y_pred_train_en = clf_en.predict(X_train)
y_pred_train_en
print('Training-set accuracy score: {0:0.4f}'. format(accuracy_score(y_train,
y_pred_train_en)))
```

```
Model accuracy score with criterion entropy: 0.8201
Training-set accuracy score: 0.9048
```

Figure 7: Accuracy of the model (Entropy criterion)

Accordingly, we will build a graph of the decision tree on Fig. 9.

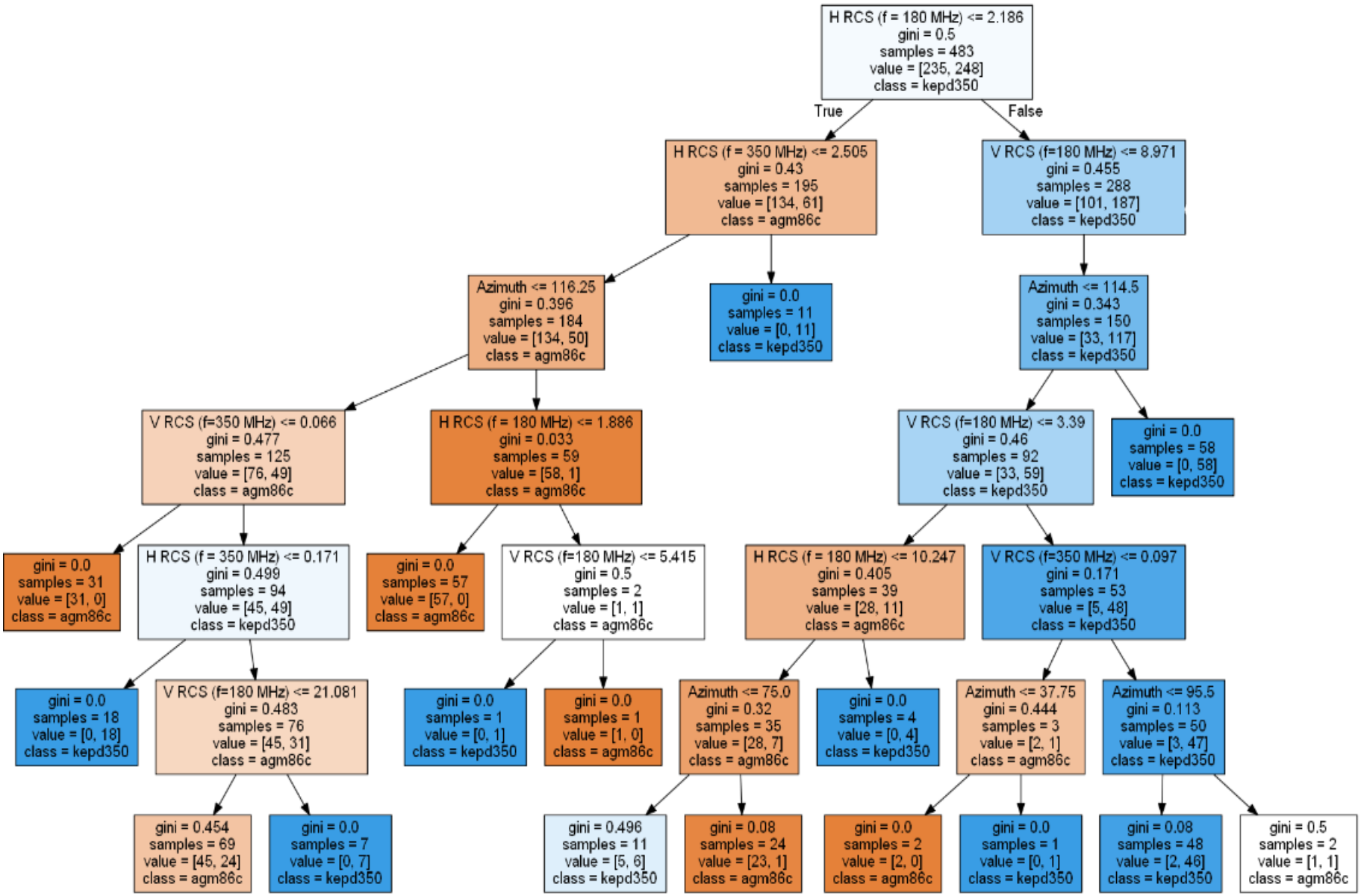

Figure 8: Decision tree graph (Gini criterion)

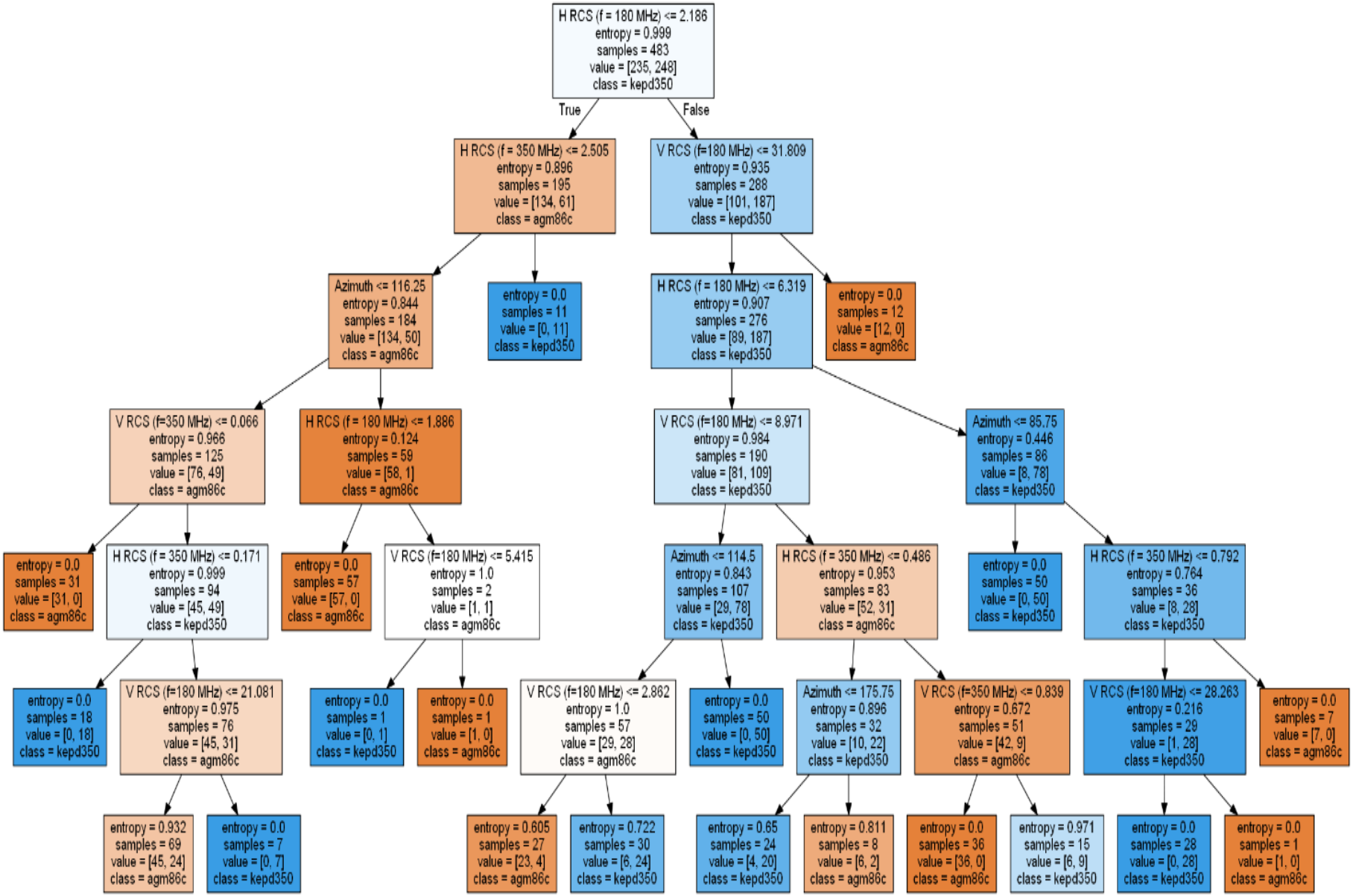

Figure 9: Decision tree graph (Entropy)

We will build a matrix of inconsistencies based on the result of the last model:

```
from sklearn.metrics import confusion_matrix
cfm = confusion_matrix(y_test, y_pred_en)
from sklearn.metrics import  confusion_matrix
conf_matrix = confusion_matrix(y_true=y_test, y_pred=y_pred_en)
fig, ax = plt.subplots(figsize=(7.5, 7.5))
ax.matshow(conf_matrix, cmap=plt.cm.Blues, alpha=0.3)
for i in range(conf_matrix.shape[0]):
    for j in range(conf_matrix.shape[1]):
        ax.text(x=j, y=i,s=conf_matrix[i, j], va='center', ha='center',
size='xx-large')
plt.xlabel('Predictions', fontsize=18)
plt.ylabel('Actuals', fontsize=18)
plt.title('Confusion Matrix', fontsize=18)
plt.show()
```

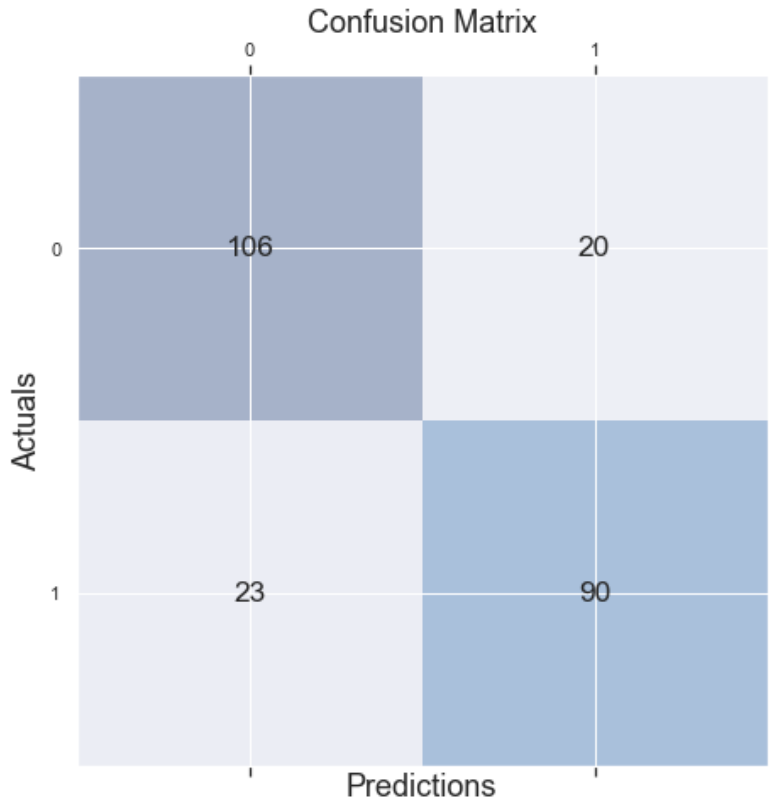

Figure 10: Matrix of inconsistencies

Finally, we will determine which criterion of the index is the most accurate:

```
if accuracy_score(y_test, y_pred_gini) > accuracy_score(y_test, y_pred_en):
       print("Gini Index Criterion is Better and it has accuracy equal to ",
accuracy_score(y_test, y_pred_gini)*100)
else:
    print("Entropy Criterion is Better and it has accuracy equal to ",
accuracy_score(y_test, y_pred_en)*100)
```

```
Gini Index Criterion is Better and it has accuracy equal to  84.51882845188284
```

Figure 11: The result of executing a block of code

5. Conclusions

No one doubts the convenience and necessity of automatic target recognition, especially nowadays. Creating new and improving old models is extremely important during wartime and preventively in peacetime. A model for recognising AGM-86C and Taurus KEPD 350 cruise missiles was built, which sufficiently accurately recognises these types of missiles (accuracy - 84.52%). The model is accurate because one of the missiles is sometimes referred to as a missile developed using "stealth" technology. The disadvantage of the model is that the model is trained on only two types of missiles. The problem is that the data on the effective scattering surface of all missiles in service of any state are extremely confidential. The decision tree method was used, which allows you to immediately highlight the advantage of the model – the clarity of the model when choosing a certain missile. A future problem of the model may be the increase in the number of missiles to be classified. In this case, it will be worth considering other methods for building a model, such as neural networks.

6. References

[1] MIT Lincoln Laboratory. Introduction to Radar Systems, 2018. URL: https://www.youtube.com/watch?v=FmW4u_AU4H4&list=PLUJAYadtuizA8RC2Qk8LfmiWA56HZsk9y&index=11.

[2] Christian Wolff. Radar Cross-Section, 2021. URL: https://www.radartutorial.eu/01.basics/Radar%20Cross%20Section.en.html.

[3] M. Skolnik, Radar Handbook, McGraw-Hill Professional, 2008. URL: https://www.accessengineeringlibrary.com/content/book/9780071485470.

[4] C. Li, Z. Peng, T. Y. Huang, T. Fan, F. K. Wang, T. S. Horng, J.-M. Muñoz-Ferreras, R. Gómez-García, L. Ran, J. Lin, A review on recent progress of portable short-range noncontact microwave radar systems, in: Proceedings of the IEEE Transactions on Microwave Theory and Techniques, 65(5), 2017, pp. 1692-1706. doi: 10.1109/TMTT.2017.2650911.

[5] H. Rahman, Fundamental Principles of Radar. CRC Press, 2019. doi: doi.org/10.1201/9780429279478.

[6] H. Zhu, Q. Li, Target Classification by Conventional Radar Based on Bispectrum and Deep CNN, Progress In Electromagnetics Research 130, (2023) 127-138. doi:10.2528/PIERC22102401.

[7] V. Shvaliuchynskyi, The Experience of the Ukrainian Army in Countering Russian Aggression, National security and the future 24(1) (2023) 83-90. doi: 10.37458/nstf.24.1.8.

[8] S. Lubiejewski, The attack-reconnaissance squadron as a new formula and a new quality of the attack helicopter squadron of the Polish Armed Forces, Security and Defence Quarterly, 2023. doi: 10.35467/sdq/159101. URL: https://securityanddefence.pl/The-attack-reconnaissance-squadron-as-a-new-formula-and-a-new-quality-of-the-attack,159101,0,2.html.

[9] J. Hunziker, E. C. Slob, J. Irving, Fast 3D ground penetrating radar simulations for glaciers, Computers & Geosciences, 173 (2023) 105320. doi: 10.1016/j.cageo.2023.105320. URL: https://www.sciencedirect.com/science/article/pii/S0098300423000249.

[10] P. J. Geller, Fielding the Long-Range Standoff Weapon Prevents a Dangerous Gap in the US Nuclear Deterrent, 2021. URL: https://www.heritage.org/sites/default/files/2021-01/BG3580.pdf.

[11] R. J. Fowler IV, Space-Based Countermeasure for Hypersonic Glide Vehicle, Ph.D. thesis, Old Dominion University, 2020. URL: https://search.proquest.com/openview/7af8527d16f70fa01b4e595d6253380e/1?pq-origsite=gscholar&cbl=18750&diss=y.

[12] S. Majumdar, The Rafale–for all Reasons, Vayu Aerospace and Defence Review 5 (2020) 32-35. URL: https://www.proquest.com/openview/ae1c5df128ff0e34d80e707f37ee1c2d/1?pq-origsite=gscholar&cbl=2028820.

[13] F. Hoffmann, Cruise missile proliferation: Trends, strategic implications, and counterproliferation, Global Security Report. Building Better Security for Wider Europe. European Leadership Network, 2021. URL: https://www.europeanleadershipnetwork.org/wp-content/uploads/2021/03/Fabian_Final-2.pdf.

[14] J. H. Pollack, C. Varriale, T. Plant, The changing role of allied conventional precision-strike capabilities in nuclear decision making, The Nonproliferation Review 27(1-3) 2020 21-37. doi: 10.1080/10736700.2020.2003561.
[15] Z. Wen, Z. Liu, S. Zhang, Q. Pan, Rotation awareness based self-supervised learning for SAR target recognition with limited training samples, in: Proceedings of the IEEE Transactions on Image Processing 30, 2021, pp. 7266-7279. doi: 10.1109/TIP.2021.3104179.
[16] X. Zhang, W. Wang, X. Zheng, Y. Wei, A novel radar target recognition method for open and imbalanced high-resolution range profile, Digital Signal Processing 118 (2021) 103212. doi: 10.1016/j.dsp.2021.103212.
[17] W. Chen, B. Chen, X. Peng, J. Liu, Y. Yang, H. Zhang, H. Liu, Tensor RNN with Bayesian nonparametric mixture for radar HRRP modeling and target recognition, in: Proceedings of the IEEE Transactions on Signal Processing, 69, 2021, pp. 1995-2009. doi: 10.1109/TSP.2021.3065847.
[18] O. Kechagias-Stamatis, N. Aouf, Automatic target recognition on synthetic aperture radar imagery: A survey, Aerospace and Electronic Systems Magazine 36(3) (2021) 56-81.
[19] B. Feng, B. Chen, H. Liu, Radar HRRP target recognition with deep networks, Pattern Recognition 61 (2017) 379-393.
[20] C. Du, B. Chen, B. Xu, D. Guo, H. Liu, Factorized discriminative conditional variational auto-encoder for radar HRRP target recognition, Signal Processing 158, (2019) 176-189. doi: 10.1016/j.sigpro.2019.01.006.
[21] F. Gao, T. Huang, J. Wang, J. Sun, A. Hussain, H. Zhou, A novel multi-input bidirectional LSTM and HMM based approach for target recognition from multi-domain radar range profiles, Electronics 8(5) (2019) 535. doi: 10.3390/electronics8050535.
[22] S. Tian, Y. Lin, W. Gao, H. Zhang, C. Wang, A multi-scale u-shaped convolution auto-encoder based on pyramid pooling module for object recognition in synthetic aperture radar images, Sensors 20(5) (2020) 1533. doi: 10.3390/s20051533.
[23] W. Jiang, Y. Ren, Y. Liu, J. Leng, A method of radar target detection based on convolutional neural network, Neural Computing and Applications 33 (2021) 9835-9847.
[24] C. Guo, Y. He, H. Wang, T. Jian, S. Sun, Radar HRRP target recognition based on deep one-dimensional residual-inception network, in: Proceedings of the IEEE Access 7 (2019) 9191-9204.
[25] A. Vasyliuk, T. Basyuk, V. Lytvyn, Specialized Interactive Methods for Using Data on Radar Application Models, CEUR Workshop Proceedings 2631 (2020) 1-11.
[26] O. Klymovych, V. Hrabchak, O. Lavrut, T. Lavrut, V. Lytvyn, V. Vysotska, The Diagnostics Methods for Modern Communication Tools in the Armed Forces of Ukraine Based on Neural Network Approach, CEUR Workshop Proceedings 2631 (2020) 198-208.
[27] S. Tyshko, O. Lavrut, V. Vysotska, O. Markiv, O. Zabula, Y. Chernichenko, T. Lavrut, Compensatory Method for Measuring Phase Shift Using Signals Bisemiperiodic Conversion in Diagnostic Intelligence Systems, CEUR Workshop Proceedings 3312 (2022) 144-154.
[28] V. Korolov, O. Korolova, I.-P. Milkovych, Y. Zaiets, V. Zhyvchuk, H. Batyshcheva, V. Lytvyn, M. Bublyk, Functional Requirements for Decision Support System of Ground Forces Recognition on Battlefield According to NATO Standards, CEUR Workshop Proceedings 2917 (2021) 503-513.
[29] I. Garkusha, V. Hnatushenko, V. Vasyliev, Research of influence of atmosphere and humidity on the data of radar imaging by Sentinel-1, in: Proceedings of the IEEE 37th International Conference on Electronics and Nanotechnology, 2017, pp. 405-408. doi: 10.1109/ELNANO.2017.7939787.
[30] A dataset for dim target detection and tracking of aircraft in radar echo sequences https://www.scidb.cn/en/detail?dataSetId=720626420979597312
[31] R. Culberg, Radar Processing, 2019. URL: https://www.kaggle.com/code/rtculberg/radar-processing/notebook.
[32] M. Kim, Radar chart, 2018. URL: https://www.kaggle.com/code/mikeskim/radar-chart/script.
[33] V. Shanbhag, Radar Traffic Data, 2020. URL: https://www.kaggle.com/datasets/vinayshanbhag/radar-traffic-data.