Issue 15, 2024

https://doi.org/10.23939/sisn2024.15.260

UDC 303.711:[531.56:623.4]

EVALUATION OF CLASSIFICATION ACCURACY USING FEEDFORWARD NEURAL NETWORK FOR DYNAMIC OBJECTS

Ivan Peleshchak[1], Diana Koshtura[2]

Lviv Polytechnic National University,
Department of Information Systems and Networks, Lviv, Ukraine
[1] E-mail: ivan.r.peleshchak@lpnu.ua, ORCID: 0000-0002-7481-8628
[2] E-mail: diana.a.koshtura@lpnu.ua, ORCID: 0000-0002-5665-5423

Abstract. This paper investigates the impact of the number of hidden layers, the number of neurons in these layers, and the types of activation functions on the accuracy of classifying projectiles of six types (A – (artillery); A/M – (artillery/missile); A/R – (armor-piercing); A/RC – (armor-piercing-incendiary); M – (missile); R – (armor-piercing shells)) using a multi-layer neural network, evaluated by a confusion matrix. Specifically, confusion matrices were constructed to assess the accuracy of classifying projectiles of six types (A – (artillery); A/M – (artillery/missile); A/R – (armor-piercing),; A/RC – (armor-piercing-incendiary); M – (missile); R – (armor-piercing shells)) using a multi-layer perceptron with one, two, and three hidden layers and activation functions: Logistic, Tanh, Relu, Softmax, respectively. It was found that the highest accuracy in classifying projectiles is achieved using a neural network with two hidden layers, with 33 neurons in the first hidden layer with Tanh activation function and 8 neurons with Tanh activation function in the second hidden layer, and Softmax for the neurons in the output layer.

Key words: confusion matrix; projectile classification; neural network; computational optimization.

Introduction

In the modern dimension of security and defense, the important task is the classification of projectiles by their type in order to ensure the safety of citizens and effective management of military operations. Various machine learning methods, including the analysis of the confusion matrix, are used to achieve this goal. The confusion matrix is a powerful tool for evaluating the effectiveness of classifiers in tasks such as projectile type classification. This article aims to conduct a comparative analysis between different configurations of feedforward neural networks in the context of projectile classification.

The task of neural network classification of projectiles (*A* – (artillery), *A/M* – (anti-tank), *A/R* – (armor-piercing), *A/RC* – (armor-piercing-incendiary), *M* – (mines), *R* – (rockets)) is relevant, as its solution automatically solves the problem of determining the type of projectile [1] (artillery or rocket) for the selection of effective strategies and tactics in military operations, for the safe removal and disposal of unexploded projectiles, and for researching the properties and characteristics of new types of projectiles.

The article considers the application of the confusion matrix for the classification of projectiles (*A* – (artillery), *A/M* – (anti-tank), *A/R* – (armor-piercing), *A/RC* – (armor-piercing-incendiary), *M* – (mines), *R* – (rockets)) using *MLP* perceptron to determine the type of projectile (artillery or rocket) using neural networks with one-, two-, three-hidden layers, having respectively the number of neurons 33; 33 and 8; 33, 16 and 8 with activation functions relu, tanh, logistic.

The work examines neural networks with the following morphologies: MLPClassifier_1HL_L (neural network with an input sensory layer of 18 neurons, a hidden layer of 33 neurons with activation functions Logistic, an output layer of 6 neurons with activation functions Softmax), MLPClassifier_2HL_T (neural network with an input sensory layer of 18 neurons, the first hidden layer of 33 neurons with activation functions Tanh, the second hidden layer of 8 neurons with activation functions Tanh, an output layer of 6 neurons with activation functions Softmax), MLPClassifier_3HL_T (neural network with an input sensory layer of 18 neurons, the first hidden layer of 33 neurons with activation functions Tanh, the second hidden layer of 8 neurons with activation functions Tanh, the third hidden layer of 16 neurons with activation functions Tanh, an output layer of 6 neurons with activation functions Softmax), and MLPClassifier_2HL_R (neural network with an input sensory layer of 18 neurons, the first hidden layer of 33 neurons with activation functions Relu, the second hidden layer of 8 neurons with activation functions Relu, an output layer of 6 neurons with activation functions Softmax).

MLPClassifier_1HL_L is a neural network with an input sensory layer of 18 neurons, which corresponds to the 18 components of the input feature vector; a hidden layer with 33 neurons with activation functions Logistics; an output layer of 6 neurons with activation functions Softmax, as 6 objects are classified.

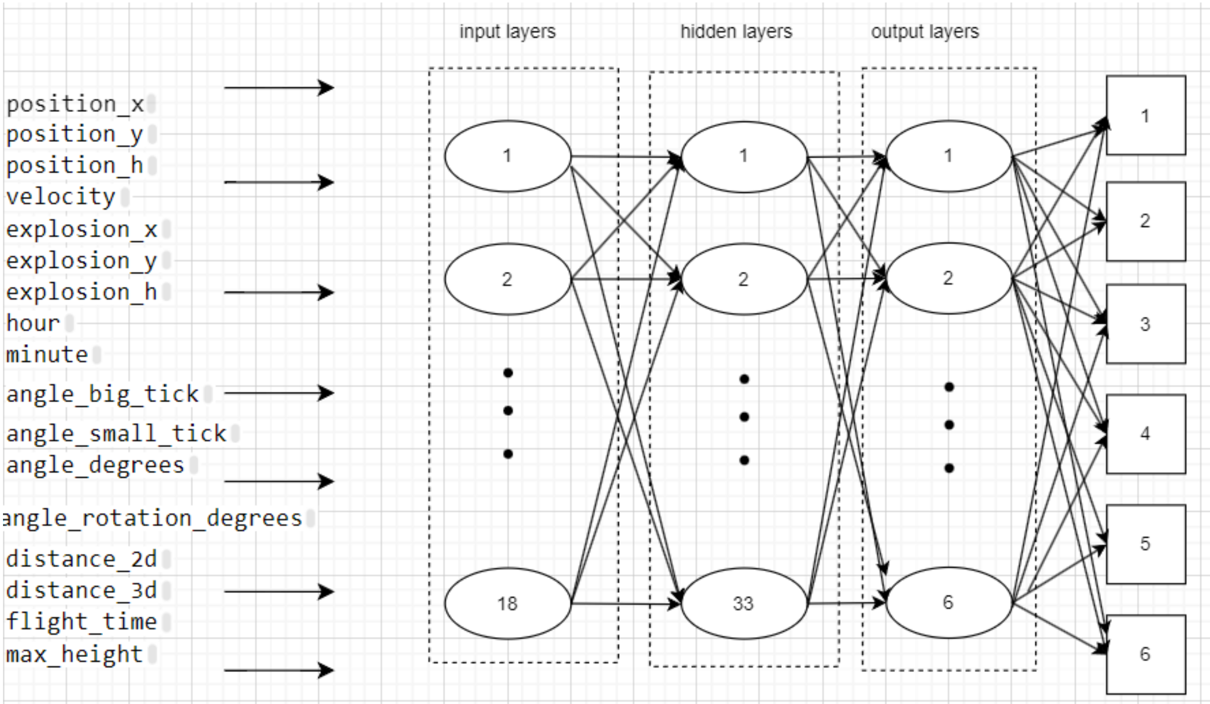

Fig. 1. Model MLPClassifier_1HL_L (formula (1))

$$y_i = f_{softmax}\left(\left(\sum_{j=1}^{33} w_{ij}\, f_{logistic}\left(\sum_{k=1}^{18} w_{jk}\, x_k\right)\right)\right)\ i \in \{1;6\}, \quad (1)$$

The architecture of the MLPClassifier_2HL_T neural network with two hidden layers and the Tanh activation function will be discussed in detail in the following sections, as this model was chosen for projectile classification due to its highest classification accuracy. The next model, MLPClassifier_3HL_T, is a neural network with an input sensory layer of 18 neurons, corresponding to the 18 components of the input feature vector; a first hidden layer of 33 neurons with activation functions Tanh; a second hidden layer of 16 neurons with activation functions Tanh; a third hidden layer of 8 neurons with activation functions Tanh; and an output layer of 6 neurons with activation functions Softmax, as 6 objects are classified.

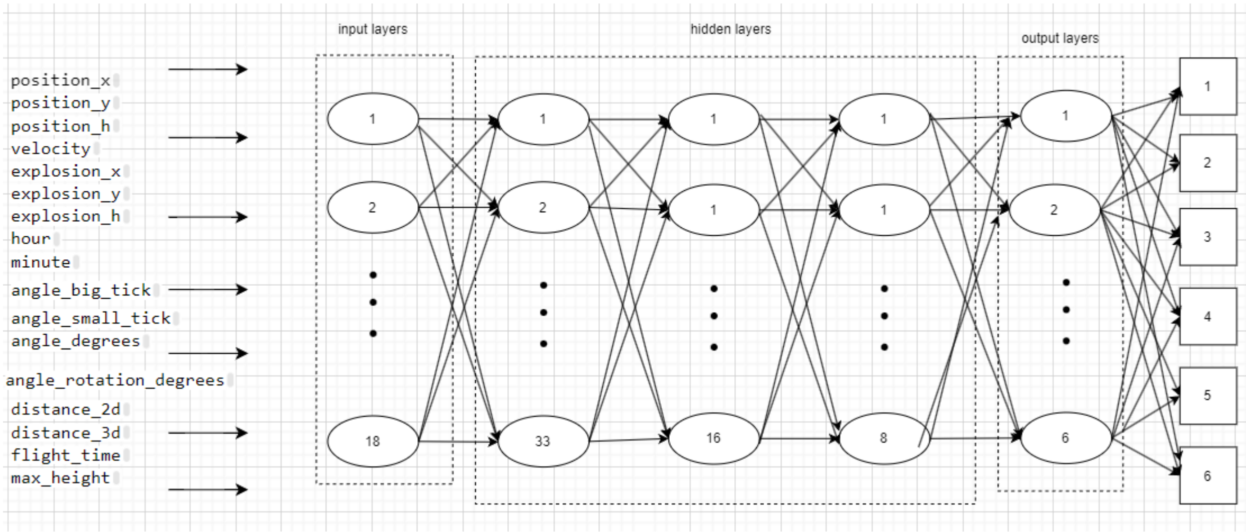

Fig. 2. Model MLPClassifier_3HL_T (formula (2))

$$y_i = f_{softmax}\left(\sum_{l=1}^{16} w_{ml} f_{tanh}\left(\sum_{i=1}^{8} w_{li} f_{tanh}\left(\sum_{j=1}^{33} w_{ij} f_{tanh}\left(\sum_{k=1}^{18} w_{jk} x_k\right)\right)\right)\right) i \in \{1; 6\}, \qquad (2)$$

It has been shown that the neural network with two hidden layers and 33 and 8 neurons with Tanh activation functions respectively, and an output layer with six neurons with a Softmax activation function, achieves the highest classification accuracy for projectiles (95.9%).

Analysis of recent research and publications

Recent publications in the field of machine learning and neural networks have demonstrated significant advancements in various application domains. Two notable areas of focus include the optimization of neural network models and the utilization of deep learning techniques for classification tasks.

In recent studies, the optimization of neural network models has been a key area of exploration. One approach introduced is the TWD-SFNN model, which integrates the three-way decisions method with feedforward neural networks. This model dynamically determines the number of hidden layer nodes and addresses challenges such as dataset discretization and learning process adjustment.

Additionally, the application of deep neural networks (DNNs) for classification tasks has garnered attention. A recent publication explores the use of DNNs for classifying space objects using light curve data. This study compares DNN-based algorithms with traditional methods and provides insights into their effectiveness and limitations.

Recent research and publications in the field of classifier performance evaluation using the confusion matrix and other metrics include various approaches and methods to improve this process. Below are several current research directions with references to relevant articles:

1. Methods for evaluating multi-label classifiers. In the article [1], the authors discuss methods for evaluating the effectiveness of classifiers for tasks where each element can belong to multiple classes simultaneously.

2. Extension of the confusion matrix. In the work [2], the authors investigate the extension of the confusion matrix to account for additional aspects of classification and uncertainty.

3. Use of a three-way confusion matrix. In the article [2], the authors use a three-way confusion matrix to assess uncertainty in decision-making.

4. Generalization of performance metrics. In the work [3], the authors generalize performance metrics for classifiers to account for the characteristics of the classification task.

5. Application of machine learning. In the article [3], researchers use machine learning methods to improve the evaluation of classifier performance.

Overall, these studies demonstrate the continuous development and improvement of methods for evaluating classifier performance for more accurate and reliable analysis of classification results in various application areas.

Formulation of the purpose of the article

The aim of the work is to evaluate the accuracy of projectile classification (A – (artillery), A/M – (anti-tank), A/R – (armor-piercing), A/RC – (armor-piercing-incendiary), M – (mines), R – (rockets)) using a neural network perceptron with different numbers of hidden layers, neurons in them, and different activation functions based on the confusion matrix.

Presentation of the main material

Let X – the set of descriptions of dynamic objects (projectiles) consists of 6 types, i.e., $|Y| = 6$. Each object x $\in X$ is described by a feature vector of 18 components. The target dependency $y^{*}: X \rightarrow Y$ is known only in a finite training sample $X^{m} = \{(x_1, y_1), \dots, (x_m, y_m)\}$, where $x_i \in R^{18}$ – is a feature vector, $y_i \in \{1,2,3,4,5,6\}$ – is the class number (type of projectile). The task is to construct an algorithm $a: X \rightarrow Y$, capable of classifying any object x $\in X$ from the 6 projectile classes with high accuracy using an optimized multilayer perceptron (with the number of hidden layers, the number of neurons in them, and the type of activation functions) based on a high-quality training sample.

For the classification of dynamic objects (A – 1, A/M – 2, A/R – 3, A/RC – 4, M – 5, R – 6), the study uses a three-layer neural network optimized by the number of hidden layers and the number of neurons in them, with the following morphology: two hidden layers and one output layer. The input sensory layer has 18 neurons, because the input feature vector has 18 components; the first hidden layer has 33 neurons with Tanh activation functions; the second hidden layer has 8 neurons with Tanh activation functions; the output layer has 6 neurons with Softmax activation functions, because there are 6 objects being classified.

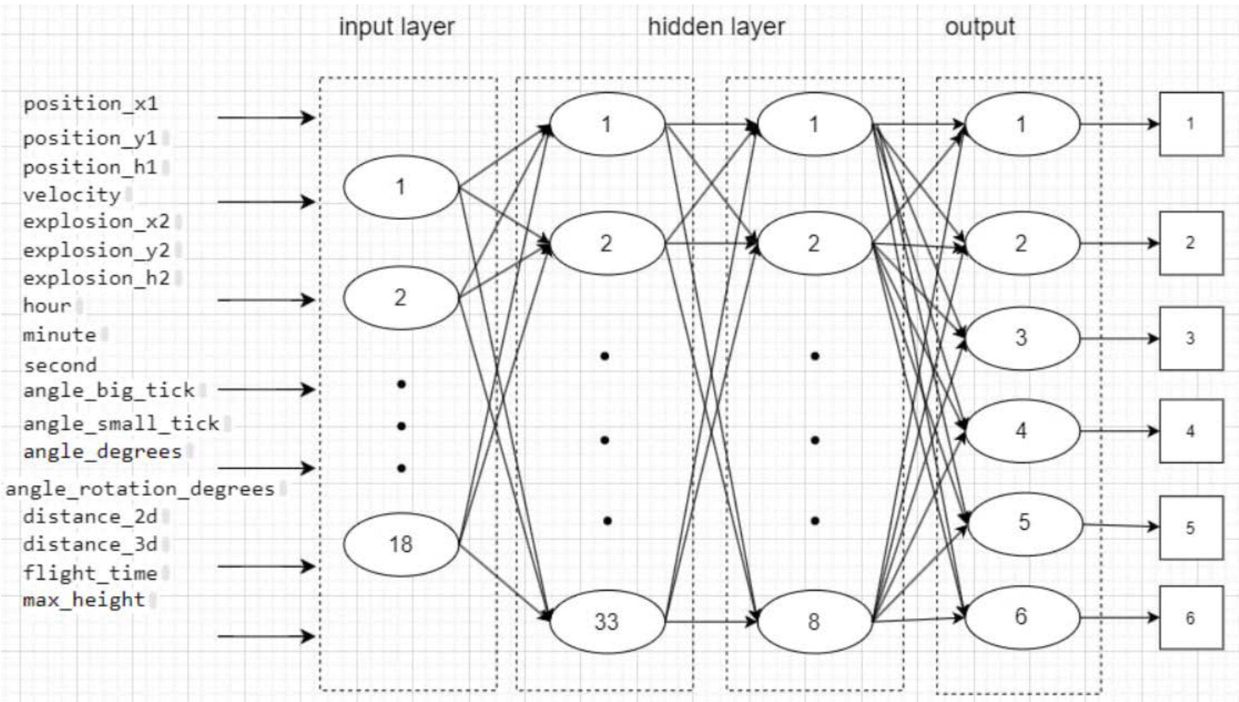

Fig. 3: Three-layer neural network model (formula(3))

Formula of the model:

$$y_i = f_{softmax}\left(\sum_{i=1}^{8} w_{li}\, f_{tanh}\left(\sum_{j=1}^{33} w_{ij}\, f_{tanh}\left(\sum_{k=1}^{18} w_{jk}\, x_k\right)\right)\right)\ i \in \{1; 6\}, \qquad (3)$$

where y_i – is an element of the output vector of probabilities for each class of projectiles; $f_{softmax}$ – is the Softmax activation function; f_{tanh} – is the Tanh activation function; w_{li} – is an element of the weight matrix between the second hidden layer and the output layer; w_{ij} – is an element of the weight matrix between the first and second hidden layers; w_{jk} – is an element of the weight matrix between the input layer and the first hidden layer, the model receives the input feature vector $\bar{x} = (x_1, x_2, \dots, x_{18})$, where x_1 – position_x_1, x_2 – position_y$_1$, x_3 – position_h_1, x_4 – velocity, x_5 – taget_class, x_6 – explosion_x_2, x_7 – explosion_y$_2$, x_8 – explosion_h_2, x_9 – hour, x_{10} – minute, x_{11} – second, x_{12} – angle_big_tick, x_{13} – angle_small_tick, x_{14} – angle_degrees, x_{15} – angle_rotation_degrees, x_{16} – distance_2d, x_{17} – distance_3d, x_{18} – flight_time.

To prepare data for the classification of projectiles from the 1L220U-KS complex, the method of pseudorandom noise with a normal distribution is used to generate new values that help the MLPClassifier_2HL_T neural network acquire generalizing properties.

Generation of a new value (formula3)):

$$V_i^* = V_i + \sigma_i \cdot N(0,1), \qquad (4)$$

where V_i^*– is the new value of the *i*-th characteristic; V_i – is the initial value of the *i*-th characteristic; σ_i – is the standard deviation of the *i*-th characteristic; $N(0,1)$ – is the function for generating pseudorandom values with a normal distribution (mean = 0, standard deviation = 1).

Calculation of the standard deviation (formula (4)):

$$\sigma_i = \sqrt{\frac{\sum (V_i - \mu_i)^2}{N}}, \qquad (5)$$

where V_i – *i*-th value of the characteristic; μ_i– mean value of the *i*-th characteristic; N – size of the training sample.

In this article, data were generated for the classification of projectiles based on data from the device 1L220U: position_x$_1$, position_y$_1$, position_h$_1$, velocity, taget_class, explosion_x$_2$, explosion_y$_2$, explosion_h$_2$, hour, minute, second, angle_big_tick, angle_small_tick, angle_degrees, angle_rotation_degrees, distance_2d, distance_3d, flight_from the study [16].

Normalization (formula(6)):

$$V' = \frac{V_i - \mu_i}{\sigma_i}, \qquad (6)$$

where V' – is the normalized value of the *i*-th characteristic; V_i – is the initial value of the *i*-th characteristic, μ_i – is the mean value of the *i*-th characteristic, σ_i is the standard deviation of the *i*-th characteristic.

The Confusion Matrix is a tool in machine learning used to evaluate the performance of a classifier on a dataset. It shows how many elements of each class were correctly or incorrectly classified by the model.

The number of rows and columns depends on the number of labels the model is supposed to predict. In this article, the classification of 6 classes is considered, so the confusion matrix for each model will have 6 rows and 6 columns (Table 1). Each row represents instances in the actual class, and each column represents instances in the predicted class.

Table 1

The template of the confusion matrix used to construct the confusion matrix for 6 classes

	Predicted Class		
Actual Class		1 (Positive)	0 (Negative)
	1 (Positive)	TP (True positive)	FN (False negative)
	0 (Negative)	FP (False positive)	TN (True negative)

FN (formula (19)–(30)): the number of falsely negative classified elements for the class will be the sum of the values of the corresponding rows, except for the TP value. FP (formula (31)–(42)): the number of falsely positive classified elements for the class will be the sum of the values of the corresponding column, except for the TP value. TN (formula (43)–(48)): the number of correctly negative classified elements for the class will be the sum of all column and row values, except for the values of this class, for which we calculate the value. TP (formula (7)–(18)): the number of correctly positive classified elements, where the actual value and the predicted value match (Table 2).

Table 2

Confusion matrix for 6 classes

A	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}
A/M	X_{21}	X_{22}	X_{23}	X_{24}	X_{25}	X_{26}
A/P	X_{31}	X_{32}	X_{33}	X_{34}	X_{35}	X_{36}
A/PC	X_{41}	X_{42}	X_{43}	X_{44}	X_{45}	X_{46}
M	X_{51}	X_{52}	X_{53}	X_{54}	X_{55}	X_{56}
P	X_{61}	X_{62}	X_{63}	X_{64}	X_{65}	X_{66}
	A	*A/M*	*A/P*	*A/PC*	*M*	*P*

The confusion matrix analyzes several neural network models in this work, so a detailed analysis of the confusion matrix will be conducted for the neural network with two hidden layers and the number of neurons 33 and 8 with Tanh activation functions, respectively, and an output layer with six neurons with a Softmax activation function, because this model showed the best results.

TP for class *A*:

$$TP_A = X_{11}, \tag{7}$$

$$TP_A = 640, \tag{8}$$

Where *TP* for class *A/M*: True Positive, meaning that the actual value and the predicted value match.

$$TP_{\frac{A}{M}} = X_{22}, \tag{9}$$

$$TP_A = 112,, \tag{10}$$

TP for class *A/P*:

$$TP_{\frac{A}{P}} = X_{33}, \tag{12}$$

$$TP_{\frac{A}{P}} = 96, \tag{12a}$$

TP for class *A/PC*:

$$TP_{\frac{A}{PC}} = X_{44}, \tag{13}$$

$$TP_{\frac{A}{PC}} = 361, \tag{14}$$

TP for class *M*:

$$TP_M = X_{55}, \tag{15}$$

$$TP_M = 58, \tag{16}$$

TP for class *P*:

$$TP_P = X_{33}, \tag{17}$$

$$TP_P = 241, \tag{18}$$

FN for class *A:*

$$FN_A = X_{12} + X_{13} + X_{14} + X_{15} + X_{16}, \tag{19}$$

$$FN_A = 1 + 0 + 6 + 0 + 0 = 7, \tag{20}$$

Where *FN* – False Negative, meaning that the actual value is positive, but it was incorrectly predicted as negative.

FN for class *A/M*:

$$FN_{\frac{A}{M}} = X_{21} + X_{23} + X_{24} + X_{25} + X_{26}, \quad (21)$$

$$FN_{\frac{A}{M}} = 1 + 0 + 0 + 1 + 0 = 2, \quad (22)$$

FN for class *A/P*:

$$FN_{\frac{A}{P}} = X_{31} + X_{32} + X_{34} + X_{35} + X_{36}, \quad (23)$$

$$FN_{\frac{A}{P}} = 0 + 0 + 1 + 0 + 6 = 7, \quad (24)$$

FN for class *A/PC*:

$$FN_{\frac{A}{PC}} = X_{41} + X_{42} + X_{43} + X_{45} + X_{46}, \quad (25)$$

$$FN_{\frac{A}{PC}} = 4 + 0 + 5 + 0 + 2 = 11, \quad (26)$$

FN for class *M*:

$$FN_M = X_{51} + X_{52} + X_{53} + X_{54} + X_{56}, \quad (27)$$

$$FN_M = 0 + 3 + 0 + 0 + 0 = 3, \quad (28)$$

FN for class *P*:

$$FN_P = X_{61} + X_{62} + X_{63} + X_{64} + X_{65}, \quad (29)$$

$$FN_P = 0 + 0 + 6 + 0 + 0 = 6, \quad (30)$$

FP for class *A:*

$$FP_A = X_{21} + X_{31} + X_{41} + X_{51} + X_{61}, \quad (31)$$

$$FP_A = 1 + 0 + 4 + 0 + 0 = 5, \quad (32)$$

Where *FP* – False Positive, meaning that the actual value is negative, but it was incorrectly predicted as positive

FP for class *A/*M:

$$FP_{\frac{A}{M}} = X_{12} + X_{32} + X_{42} + X_{52} + X_{62}, \quad (33)$$

$$FP_{\frac{A}{M}} = 1 + 0 + 0 + 3 + 0 = 4, \quad (34)$$

FP for class *A/P*:

$$FP_{\frac{A}{P}} = X_{13} + X_{23} + X_{43} + X_{53} + X_{63}, \quad (35)$$

$$FP_A = 0 + 0 + 5 + 0 + 6 = 11, \quad (36)$$

FP for class *A/PC*:

$$FP_{\frac{A}{PC}} = X_{14} + X_{24} + X_{34} + X_{54} + X_{61}, \quad (37)$$

$$FP_{\frac{A}{PC}} = 6 + 0 + 1 + 0 + 0 = 7, \quad (38)$$

FP for class M:

$$FP_M = X_{15} + X_{25} + X_{35} + X_{45} + X_{65,} \quad (39)$$

$$FP_M = 0 + 1 + 0 + 0 + 0 = 1, \quad (40)$$

FP for class *P*:

$$FP_P = X_{16} + X_{26} + X_{36} + X_{46} + X_{56}, \quad (41)$$

$$FP_P = 0 + 0 + 6 + 2 + 0 = 8, \quad (42)$$

TN (True Negative) – the sum of values in all other rows and columns except for the class for which we are calculating the value

$$TN_A = 112 + 1 + 96 + 1 + 6 + 5 + 361 + 2 + 3 + 58 + 6 + 241 = 892, \quad (43)$$

$$TN_{\frac{A}{M}} = 640 + 6 + 1 + 4 + 5 + 361 + 2 + 58 + 6 + 241 = 1324, \quad (44)$$

$$TN_{\frac{A}{P}} = 640 + 1 + 6 + 1 + 112 + 1 + 4 + 361 + 2 + 3 + 58 + 241 = 1430, \quad (45)$$

$$TN_{\frac{A}{PC}} = 640 + 1 + 112 + 1 + 96 + 3 + 58 + 6 + 241 = 1158, \quad (46)$$

$$TN_M = 640 + 1 + 6 + 1 + 112 + 96 + 1 + 6 + 4 + 5 + \ 361 + 6 + 241 = 1420, \tag{47}$$

$$TN_P = 640 + 1 + 6 + 1 + 112 + 1 + 96 + 1 + 4 + 5 + 361 + 3 + 358 + 3 + 58 = 1650, \tag{48}$$

In the top left corner of the confusion matrix are the names of the actual classes. In the bottom part of the confusion matrix are the names of the predicted classes.

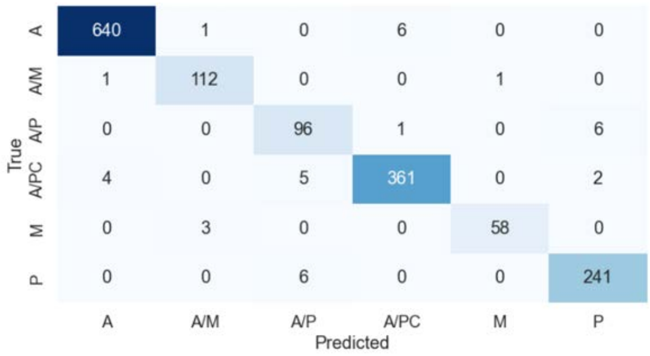

Fig. 4. Confusion matrix for MLPClassifier_2HL_T training dataset

After analyzing the confusion matrix, the following conclusions can be drawn for the training dataset:

1. Classification effectiveness:
- The model showed high effectiveness in classifying certain classes, particularly "*A*" and "*A/PC*," where the *TP* values are high: 640 and 361 respectively.
- However, for some other classes, such as "*A/P*" and "*M*," the model exhibited less satisfactory results with a noticeable number of *FP* and *FN* values.

2. Misclassification:
- There is a certain amount of misclassified samples in all classes, indicating the model's inability to correctly identify certain subclasses.
- The values of *FP* and *FN* for some classes indicate that the model tends to incorrectly classify some classes as others.

3. Class balance:
- Considering the *TN* values, it can be understood that the model successfully identifies classes that do not belong to a specific class, except for the class for which the value is being calculated.

4. Need for further improvement:
- Focusing on improving the classification accuracy for classes with a high number of errors (FP and FN) may enhance the overall effectiveness of the model.

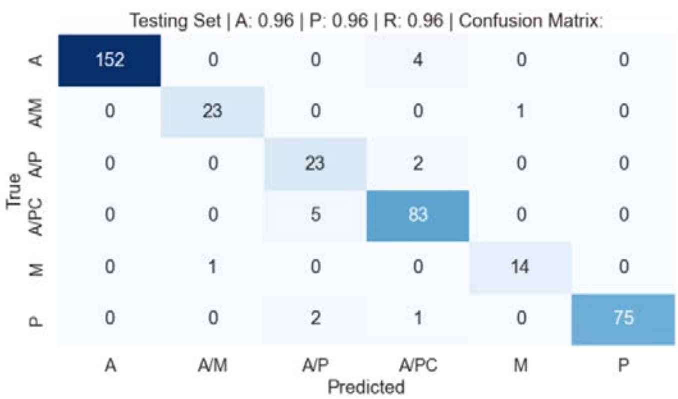

Fig. 5. Confusion matrix for MLPClassifier_2HL_T test dataset

The values of accuracy: 0.96, precision: 0.96, recall: 0.96 shown in Fig. 2 above indicate that the MLPClassifier_2HL_T model performs with high classification effectiveness on the test dataset.

The next model to be considered is a neural network with two hidden layers and 33 and 8 neurons with Relu activation functions respectively and an output layer with six neurons with Softmax activation – MLPClassifier_2HL_R.

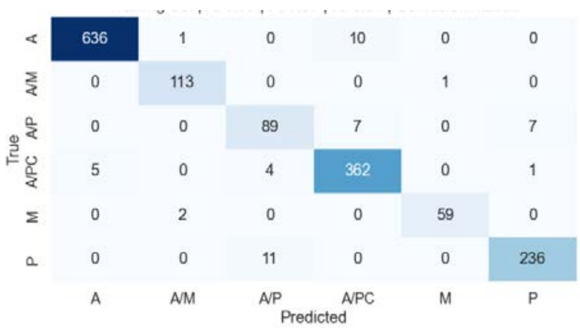

Fig. 6. Confusion matrix for MLPClassifier_2HL_R training dataset

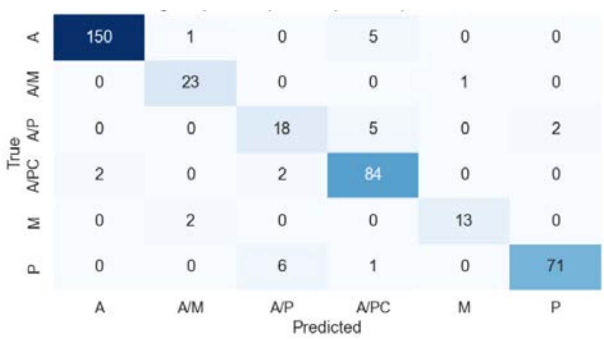

Fig. 7. Confusion matrix for MLPClassifier_2HL_R test dataset

Analysis of the confusion matrix for the training and test datasets for MLPClassifier_2HL_R allows the following conclusions to be drawn:

1. Class "*A*": MLPClassifier_2HL_R correctly classified 636 samples of class "*A/P*" in the training dataset and 150 samples in the test dataset, indicating high efficiency in classifying this class.

2. Class "*A/M*": MLPClassifier_2HL_R also showed high efficiency in classifying samples of class "*A/M*" in the training dataset, correctly classifying 113 samples, but lower efficiency in the test dataset, correctly classifying only 23.

3. Class "*A/P*": For the class "*A/P*", MLPClassifier_2HL_R correctly classified 89 samples in the training dataset and 18 samples in the test dataset, indicating moderate efficiency in classifying this class.

4. Class "*A/PC*": MLPClassifier_2HL_R correctly classified 362 samples of class "*A/PC*" in the training dataset and 84 samples in the test dataset, indicating average efficiency in classifying this class.

5. Class "*M*": The efficiency of MLPClassifier_2HL_R for class "*M*" was lower than for other classes, correctly classifying only 59 samples in the training dataset and 13 in the test dataset.

6. Class "*P*": MLPClassifier_2HL_R correctly classified 236 samples of class "*P*" in the training dataset and 71 in the test dataset, indicating high efficiency in classifying this class.

Overall, MLPClassifier_2HL_R has shown quite high efficiency in classifying some classes (e.g., "*A*" and "*A/PC*"), but for other classes (especially "*A/P*" and "*M*"), the results were unsatisfactory. This may require further refinement of the model to achieve better results on the test dataset.

Comparing the confusion matrix of MLPClassifier_2HL_R and MLPClassifier_2HL_T, MLPClassifier_2HL_R correctly classified 636 samples of class "*A*", while MLPClassifier_2HL_T classified 640 in the training dataset; and MLPClassifier_2HL_R correctly classified 150 samples of class "*A*", while MLPClassifier_2HL_T classified 152 in the test dataset, indicating high efficiency in classifying this class in both models, but MLPClassifier_2HL_T shows slightly better results.

Comparing the confusion matrices of the two models – MLPClassifier_2HL_R and MLPClassifier_2HL_T, we can see that both models demonstrate high efficiency in classifying dynamic objects, but MLPClassifier_2HL_T shows slightly better results.

The next model to be considered is a neural network with one hidden layer and 33 neurons with Relu activation function, and an output layer with six neurons with Softmax activation function – MLPClassifier_1HL_L.

Comparing the confusion matrix of MLPClassifier_1HL_L with MLPClassifier_2HL_T, we can see that MLPClassifier_1HL_L correctly classified 632 samples of class "*A*," while MLPClassifier_2HL_T classified 640 correctly on the training dataset; and MLPClassifier_1HL_L correctly classified 149 samples of class "*A*", while MLPClassifier_2HL_T classified 152 correctly on the test dataset, indicating high effectiveness in classifying this class in both models, but MLPClassifier_2HL_T shows slightly better results.

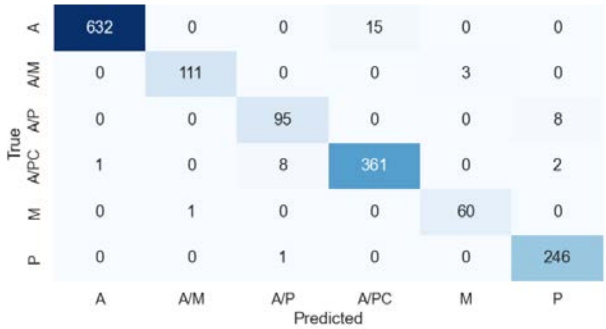

Fig. 8. Confusion Matrix for MLPClassifier_1HL_L Training Datase

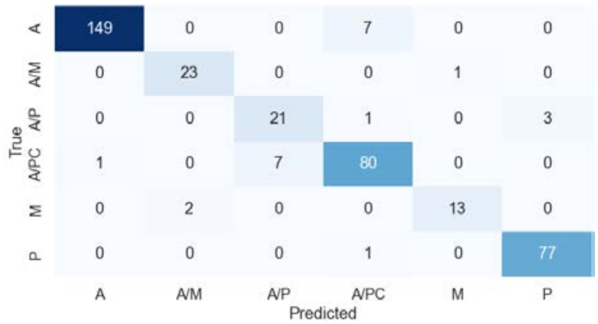

Fig. 9. Confusion Matrix for MLPClassifier_1HL_L Test Dataset

Comparing the confusion matrices of the two models – MLPClassifier_1HL_L and MLPClassifier_2HL_T, we can see that both models demonstrate high effectiveness in classifying dynamic objects, but MLPClassifier_2HL_T shows slightly better results.

The next model to be considered is a neural network with three hidden layers and 33 and 8 neurons with Tanh activation functions, respectively, and an output layer with six neurons with a Softmax activation function – MLPClassifier_3HL_T.

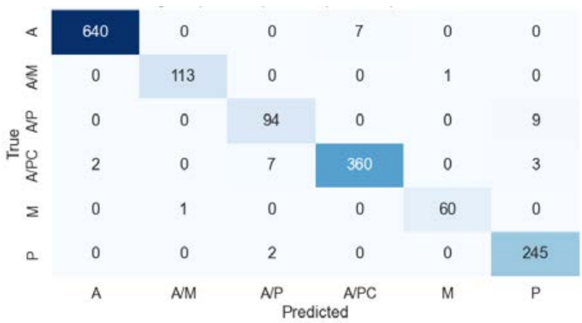

Fig. 10. Confusion Matrix for MLPClassifier_3HL_T Training Dataset

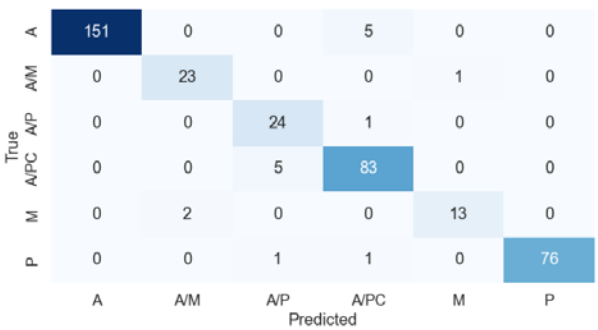

Fig. 11. Confusion Matrix for MLPClassifier_3HL_T Testing Dataset

Comparing the confusion matrix of MLPClassifier_3HL_T and MLPClassifier_2HL_T, it can be seen that MLPClassifier_3HL_T correctly classified 640 instances of class "*A*," while MLPClassifier_2HL_T classified 640 correctly in the training dataset. In addition, MLPClassifier_3HL_T correctly classified 151 instances of class "*A*", while MLPClassifier_2HL_T classified 152 correctly in the testing dataset, indicating high efficiency in classifying this class in both models, but MLPClassifier_2HL_T shows slightly better results.

Comparing the confusion matrices of the two models – MLPClassifier_3HL_T and MLPClassifier_2HL_T, both models demonstrate high efficiency in classifying dynamic objects, but MLPClassifier_2HL_T shows slightly better results. Additionally, MLPClassifier_2HL_T has two hidden layers, while MLPClassifier_3HL_H has three hidden layers, so for optimizing the computational resources of the neural network, it is advisable to use a neural network with two hidden layers.

Conclusions

Based on the analysis of the confusion matrices for the MLPClassifier models, the following detailed conclusions can be made:

1. The two-layer perceptron with Tanh activation function, MLPClassifier_2HL_T, shows high accuracy for classes "*A*" and "*A/PC*", with TP values of 640 and 361 respectively.

2. The two-layer perceptron with Relu activation function, MLPClassifier_2HL_R, shows lower classification accuracy than the MLPClassifier_2HL_T model. For classes "*A/P*" and "M," the results were satisfactory – 18 and 13 respectively. Compared to MLPClassifier_2HL_T, both models showed similar results for class "*A*", but MLPClassifier_2HL_R showed slightly worse results.

3. The perceptron with Logistic activation function, MLPClassifier_1HL_L, demonstrates high classification accuracy for class "*A*" but less satisfactory results for some other classes. Compared to MLPClassifier_2HL_T, MLPClassifier_1HL_L shows similar effectiveness in classifying class "*A*," but the results differ for other classes.

4. The three-layer perceptron with Tanh activation function, MLPClassifier_3HL_T, shows high classification accuracy for some classes, but the results for classes "*A/P*" and "M" are not satisfactory. Compared to MLPClassifier_2HL_T, MLPClassifier_3HL_T shows similar results but with greater complexity due to the third hidden layer.

Thus, the analysis based on the confusion matrix shows that the highest accuracy in classifying projectiles (A – (ammunition); *A/M* – (ammunition/explosive); *A/P* – (armor-piercing); *A/PC* – (armor-piercing-incendiary); *M* – (explosive); *P* – (armor-piercing ammunition)) is achieved using a neural network perceptron with the following morphology (formula (49)):

$$y_i = f_{softmax}\left(\sum_{i=1}^{8} w_{li}\, f_{tanh}\left(\sum_{j=1}^{33} w_{ij}\, f_{tanh}\left(\sum_{k=1}^{18} w_{jk}\, x_k\right)\right)\right)\; i \in \{1; 6\}. \qquad (49)$$

So, the input sensor layer has 18 neurons because the input feature vector has 18 components; the first hidden layer has 33 neurons with Tanh activation functions; the second hidden layer has 8 neurons with Tanh activation functions; and the output layer has 6 neurons with Softmax activation functions because there are 6 objects being classified.

Список літератури

1. Крстініч, Д., Бравіч, М., Шеріч, Л., та Божіч-Штуліч, Д. (2020). Оцінка продуктивності багатоетикеткового класифікатора з використанням матриці помилок. Комп'ютерні науки та інформаційні технології, 1, 1–14. DOI:10.5121/csit.2020.100801

2. Сю, Ж., Чжан, Ї., та Мяо, Д. (2020). Трьохстороння матриця помилок для класифікації: вимірювальний погляд. Інформаційні науки, 507, 772–794. DOI:10.1016/j.ins.2019.06.064

3. Хейдаріан, М., Дойл, Т. Е., та Самаві, Р. (2022). MLCM: Багатоетикеткова матриця помилок. IEEE Access, 10, 19083–9095. DOI:10.1109/ACCESS.2022.3151048

4. Вуєвіч, Ж. (2021). Метрики оцінки моделей класифікації. Міжнародний журнал передових комп'ютерних наук і застосувань, 12(6), 599–606. DOI:10.14569/IJACSA.2021.0120670

5. Хаснайн, М., Паша, М. Ф., Гані, І., Імран, М., Аль-Захрані, М. Й., та Будіарто, Р. (2020). Оцінка передбачення довіри та міри матриці помилок для ранжування веб-сервісів. IEEE Access, 8, 90847–90861. DOI:10.1109/ACCESS.2020.2994222

6. Чжоу, С., та Дель Валле, А. (2020, березень). Діапазонна матриця помилок для класифікації дисбалансованих часових рядів. У 2020 6-й конференції зі науки про дані та застосувань машинного навчання (CDMA) (стор. 1–6). IEEE. DOI:10.1109/CDMA47397.2020.00006

7. Санні, Р. Р., та Гурупрасад, Х. С. (2021). Аналіз метрик продуктивності пацієнтів з серцевою недостатністю за допомогою Python та алгоритмів машинного навчання. Глобальні переходи: матеріали конференцій, 2(2), 233–237. DOI:10.1016/j.gltp.2021.08.028

8. Гупта, А., Пармар, Р., Сурі, П., та Кумар, Р. (2021, грудень). Визначення точності штучних інтелектуальних моделей за допомогою Python та R-Studio. У 2021 3-й Міжнародній конференції з передових досягнень в області обчислювальної техніки, зв'язку, керування та мереж (ICAC3N) (стор. 889–894). IEEE. DOI:10.1109/ICAC3N53548.2021.9725687

9. Пелещак, Р., Литвин, В., Пелещак, І., Худий, А., Рибчак, З., та Мушаста, С. (2022). Класифікація тональності тексту за допомогою гібридної згорткової нейронної мережі з паралельними та послідовними з'єднаннями між шарами. У COLINS (стор. 904–915). DOI:10.3390/sym16040485

10. Пелещак, Р., Литвин, В., Холодна, Н., Пелещак, І., та Висоцька, В. (2022, лютий). Двоетапний метод шифрування AES на основі стохастичної помилки нейронної мережі. У 2022 IEEE 16-й Міжнародній конференції з передових тенденцій у радіоелектроніці, телекомунікаціях та комп'ютерному інженерії (TCSET) (стор. 381–385). IEEE. DOI:10.1109/AIACT.2019.8847896

11. Шамрат, Ф. Дж. М., Азам, С., Карім, А., Ахмед, К., Буі, Ф. М., та Де Бур, Ф. (2023). Високоточна багатокласова класифікація захворювань легень через налаштований MobileNetV2 за допомогою зображень рентгенів грудей. Комп'ютери в біології та медицині, 155, 106646. DOI:10.1016/j.compbiomed.2023.106646

12. Хан, М. С., Нат, Т. Д., Хоссен, М. М., Мукерджі, А., Хаснат, Х. Б., Мім, Т. М., & Хан, У. (2023). Порівняння технік багатокласової класифікації за допомогою набору даних з сухого кавового зерна. Міжнародний журнал когнітивних обчислень в інженерії, 4, 6–20. DOI:10.1016/j.ijcce.2023.01.002

13. Нахідузаман, М., Гоні, М. О. Ф., Хасан, Р., Іслам, М. Р., Сифулла, М. К., Шахріар, С. М., ... та Ковальський, М. (2023). Паралельна CNN-ELM: багатокласова класифікація зображень рентгенів грудної клітки для виявлення сімнадцяти захворювань легень, включаючи COVID-19. Експертні системи з застосуваннями, 229, 120528. DOI:10.21203/rs.3.rs-3358084/v1

14.Ду, Ї., Ян, Ї., Тао, Д., та Хсіє, М. Х. (2023). Проблемно-залежна потужність квантових нейронних мереж у багатокласовій класифікації. Фізичний огляд листів, 131(14), 140601. DOI:10.1103/PhysRevLett.131.140601

15. Афзал, С., Зіапур, Б. М., Шокрі, А., Шакібі, Х., та Собхані, Б. (2023). Прогнозування споживання енергії будівель за допомогою моделей, підтриманих багатошаровими перцептронами нейронних мереж; порівняння різних алгоритмів оптимізації. Енергія, 282, 128446. DOI:10.1016/j.energy.2023.128446

16. UkrOboronProm. (2023). Каталог радіолокаційних, радіозв'язку та систем ППО. Отримано з http://progress.gov.ua/en/catalogs/

References

1. Krstinić, D., Braović, M., Šerić, L., & Božić-Štulić, D. (2020). Multi-label classifier performance evaluation with confusion matrix. Computer Science & Information Technology, 1, 1–14. DOI:10.5121/csit.2020.100801

2. Xu, J., Zhang, Y., & Miao, D. (2020). Three-way confusion matrix for classification: A measure driven view. Information sciences, 507, 772–794. DOI:10.1016/j.ins.2019.06.064

3. Heydarian, M., Doyle, T. E., & Samavi, R. (2022). MLCM: Multi-label confusion matrix. IEEE Access, 10, 19083–19095. DOI:10.1109/ACCESS.2022.3151048

4. Vujović, Ž. (2021). Classification model evaluation metrics. International Journal of Advanced Computer Science and Applications, 12(6), 599–606. DOI:10.14569/IJACSA.2021.0120670

5. Hasnain, M., Pasha, M. F., Ghani, I., Imran, M., Alzahrani, M. Y., & Budiarto, R. (2020). Evaluating trust prediction and confusion matrix measures for web services ranking. Ieee Access, 8, 90847–90861. DOI:10.1109/ACCESS.2020.2994222

6. Zhou, X., & Del Valle, A. (2020, March). Range based confusion matrix for imbalanced time series classification. In 2020 6th Conference on Data Science and Machine Learning Applications (CDMA) (pp. 1–6). IEEE. DOI:10.1109/CDMA47397.2020.00006

7. Sanni, R. R., & Guruprasad, H. S. (2021). Analysis of performance metrics of heart failured patients using Python and machine learning algorithms. Global transitions proceedings, 2(2), 233–237. DOI:10.1016/j.gltp.2021.08.028

8. Gupta, A., Parmar, R., Suri, P., & Kumar, R. (2021, December). Determining Accuracy Rate of Artificial Intelligence Models using Python and R-Studio. In 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N) (pp. 889–894). IEEE. DOI:10.1109/ICAC3N53548.2021.9725687

9. Peleshchak, R., Lytvyn, V., Peleshchak, I., Khudyy, A., Rybchak, Z., & Mushasta, S. (2022). Text Tonality Classification Using a Hybrid Convolutional Neural Network with Parallel and Sequential Connections Between Layers. In COLINS (pp. 904–915). DOI:10.3390/sym16040485

10. Peleshchak, R., Lytvyn, V., Kholodna, N., Peleshchak, I., & Vysotska, V. (2022, February). Two-stage AES encryption method based on stochastic error of a neural network. In 2022 IEEE 16th International Conference

on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET) (pp. 381–385). IEEE. DOI:10.1109/AIACT.2019.8847896

11. Shamrat, F. J. M., Azam, S., Karim, A., Ahmed, K., Bui, F. M., & De Boer, F. (2023). High-precision multiclass classification of lung disease through customized MobileNetV2 from chest X-ray images. Computers in Biology and Medicine, 155, 106646. DOI:10.1016/j.compbiomed.2023.106646

12. Khan, M. S., Nath, T. D., Hossain, M. M., Mukherjee, A., Hasnath, H. B., Meem, T. M., & Khan, U. (2023). Comparison of multiclass classification techniques using dry bean dataset. International Journal of Cognitive Computing in Engineering, 4, 6–20. DOI:10.1016/j.ijcce.2023.01.002

13. Nahiduzzaman, M., Goni, M. O. F., Hassan, R., Islam, M. R., Syfullah, M. K., Shahriar, S. M., ... & Kowalski, M. (2023). Parallel CNN-ELM: A multiclass classification of chest X-ray images to identify seventeen lung diseases including COVID-19. Expert Systems with Applications, 229, 120528. DOI:10.21203/rs.3.rs-3358084/v1

14. Du, Y., Yang, Y., Tao, D., & Hsieh, M. H. (2023). Problem-dependent power of quantum neural networks on multiclass classification. Physical Review Letters, 131(14), 140601. DOI:10.1103/PhysRevLett.131.140601

15. Afzal, S., Ziapour, B. M., Shokri, A., Shakibi, H., & Sobhani, B. (2023). Building energy consumption prediction using multilayer perceptron neural network-assisted models; comparison of different optimization algorithms. Energy, 282, 128446. DOI:10.1016/j.energy.2023.128446

16. UkrOboronProm. (2023). Catalogue Radar, Radio Communication and Air Defence Systems. Retrieved from http://progress.gov.ua/en/catalogs/

ОЦІНКА ТОЧНОСТІ КЛАСИФІКАЦІЇ ЗА ДОПОМОГОЮ НЕЙРОМЕРЕЖІ ПРЯМОГО ПОШИРЕННЯ ДИНАМІЧНИХ ОБ’ЄКТІВ

Іван Пелещак[1], Діана Коштура[2]

Національний університет “Львівська політехніка”,
кафедра інформаційних систем та мереж, Львів, Україна
[1] E-mail: ivan.r.peleshchak@lpnu.ua, ORCID: 0000-0002-7481-8628
[2] E-mail: diana.a.koshtura@lpnu.ua, ORCID: 0000-0002-5665-5423

У роботі на основі мультишарової нейронної мережі досліджено вплив кількості прихованих шарів, числа нейронів у них та типів активаційних функцій на точність класифікації снарядів шістьох типів (A – (а-боєприпаси); A/M – (а/м-боєприпаси; A/P – (бронебійні); A/PC – (бронебійно-набивні); M – (m-боєприпаси); P – (бронебійні боєприпаси)), яка оцінюється матрицею помилок. Зокрема, побудовані матриці помилок для оцінки точності класифікації снарядів шістьох типів (A – (а-боєприпаси); A/M – (а/м-боєприпаси; A/P – (бронебійні); A/PC – (бронебійно-набивні); M – (m-боєприпаси); P – (бронебійні боєприпаси) мультишаровим нейронним перцептроном з одним, двома та трьома прихованими шарами та функціями активацій: Logistic, Tanh, Relu, Softmax відповідно. Встановлено, що найвища точність класифікації снарядів досягається за допомогою нейромережі з двома прихованими шарами з кількістю нейронів у першому прихованому шарі 33 з функцією активації tanh та 8–ми нейронами з функцією ативації Tanh у другому прихованому шарі та Softmax для нейронів вихідног шару.

Ключові слова: матриця помилок; класифікація снарядів; нейронна мережа; оптимізація обчислювальних ресурсів.