Автоматизовані системи управління та прилади автоматики. 2026. № 1 (188) *ISSN 3083-7715 (online)*
Management Information System and Devices. 2026. No. 1 (188) *ISSN 3083-7650 (0135-1710)(print)*

UDC 004.032.26; 681.5+519.711.2 DOI: https://doi.org/10.30837/0135-1710.2026.188.017

Igor Grebennik, Oleksii Kovalenko

USE OF ARTIFICIAL NEURAL NETWORKS FOR MANAGING THE SORTING OF PARCELS IN A CONVEYOR FLOW

The subject matter of the article *is the decision-making models of automated sorting lines (ASLs) installed at parcel sorting centers (PSCs).* ***The goal of the work*** *is to develop a decision-making model for the ASLs serving the loading doors of PSC terminals in the form of an ANN that classifies parcels according to their weight and dimensions.* ***The following tasks were solved in the article:*** *an analysis of the design modifications of ASL equipment for implementing three variants of truck loading logic was conducted; the weight and dimensional ranges of parcels associated with loading chute and loading door numbers were determined; the criteria for implementing the specified parcel sorting logic; the parameters of training and testing datasets for ANN models were determined; using MATLAB tools, training and testing of four types of ANNs were carried out: RBFNN, GRNN, PNN, FFNN; a comparative analysis of the ANNs was conducted using the relative precision of correct parcel classification by weight and dimensional ranges and the number of neurons employed as quality metrics.* ***Methods used:*** *systems analysis, analytical methods, computer simulation, ANN architecture training methods, and mathematical and statistical analysis of training effectiveness.* ***Results achieved.*** *Comparative training and testing of four types of ANNs was performed. Based on the test results, the FFNN model trained using the Levenberg–Marquardt method was selected for implementation, as it provides a relative precision (RP) of correct classification 97%.* ***Conclusions.*** *The developed ASL decision-making model in the form of an FFNN enables the classification of parcels within a conveyor flow into three ranges of weight and dimensional parameters, thereby implementing the specified sorting logic across loading doors or chutes. The implemented sorting logic ensures compact loading of truck cargo compartments and reduces the risk of parcel damage when parcels are stacked on top of one another.*

Keywords: *delivery logistics; decision-making model; classification; sorting line; artificial neural networks.*

1. Introduction

Statement of the problem

Industrial automation of object sorting is one of the key areas of development in modern logistics associated with the provision of postal services in the parcel delivery industry (PDI) [1] and cross-docking services [2] for delivering goods from manufacturers to the warehouses of e-commerce retailers. To support the PDI, logistics companies employ a network of intermediate and final parcel sorting centers (PSCs), central parcel consolidation terminals (CPCTs), and parcel hubs (PHs). To ensure the delivery of goods from manufacturers to consumers, a network of intermediate and final cross-docking distribution centers is used. The aforementioned transhipment nodes of delivery logistics networks are designed for unloading inbound trucks, sorting freight, and loading outbound trucks for further delivery to the addresses of subsequent network nodes.

Automated sorting lines (ASLs) are used for object sorting and serve as structural components for constructing automated sorting conveyors (ASCs). Such ASCs are installed at PSCs and cross-docking distribution centres.

A major issue in applying ASLs within the PDI is the limitation of their decision-making models, which arises from the uncertainty of parcel sorting conditions based on categories defined by weight and dimensional parameters. This problem is caused by the limited functional capabilities of ASLs, which sort parcels solely by delivery address without considering their weight and dimensions. This may lead to inefficient utilisation of truck cargo space during loading, as well as the risk of parcel damage when parcels are stacked on top of one another. This problem necessitates research into the development of an ASL decision-making model that enables parcel sorting management based on weight and dimensional classification to implement the specified outbound truck loading logic.

This article is devoted to the development of an ASL decision-making model in the form of a neural network that implements the specified classification of parcels by their weight and dimensions. This choice of model implementation is motivated by recent advances in the use of neural networks, which have led to significant progress in solving complex problems associated with virtually every field of human activity [3–5]. The objective of this study is to compare neural networks and machine learning algorithms for implementing an ASL decision-making model that classifies and sorts

© Igor Grebennik, Oleksii Kovalenko, 2026
This is an Open Access article under the CC BY-NC-SA 4.0 license https://creativecommons.org/licenses/by-nc-sa/4.0/deed.uk

ISSN 3083-7715 (online) Автоматизовані системи управління та прилади автоматики. 2026. № 1 (188)
ISSN 3083-7650 (0135-1710)(print) Management Information System and Devices. 2026. No. 1 (188)

parcels according to specified ranges of their weight and dimensional parameters.

2. Analysis of recent studies and publications

For parcel sorting, PDI transhipment centres are equipped with ASCs that incorporate ASLs. In general, the decision-making model of an ASC at a PSC is implemented as a computer system that monitors parcel unloading operations and their sorting based on barcode label information, followed by loading. An analysis of publications related to the improvement of postal service logistics within the PDI is associated with the overarching task of developing decision-making models for ASLs or ASCs to ensure the continuous (round-the-clock) and uninterrupted operation of intermediate PSCs. The problems addressed can be divided into two research directions.

The first research direction addresses the scheduling of freight transport across transhipment points of the logistics network, including arrival and departure times. In [6], a genetic algorithm was developed to minimise idle time by monitoring the parcel unloading rate to manage the dispatch schedule of outbound trucks, regardless of the loading volume of their cargo compartments. In [1, 7], a PSC model and genetic algorithms were developed to plan the arrival times of inbound trucks in order to minimise unloading time. For planning purposes, the genetic algorithm in [1] uses information from the PSC model regarding the number of available unloading doors, while the genetic algorithm in [7] uses information on the cargo volumes of inbound trucks. In [8], an adaptive genetic algorithm was developed that uses parcel label information to determine an unloading plan for inbound trucks with unequal cargo volumes. The algorithm also enables the determination of a shortened parcel transport route along the conveyor, thereby increasing the throughput of the CPCT. The study in [8] was continued in [9], where a CPCT simulation model and a hybrid genetic algorithm were developed to determine an outbound truck loading plan, taking into account the parcel transport time along shortened routes. In [10], a combinatorial algorithm was developed for planning train interval schedules for their allocation to railway tracks in order to minimise idle time at railway transhipment points.

The second direction is associated with research aimed at increasing PSC throughput, taking into account the configuration of ASC structural elements. In [11], a genetic algorithm was developed to plan the optimal structure of ASC components that ensures maximum throughput. A CPCT simulation model and parcel label information are used to verify the algorithm's performance. In [12], a mixed-integer programming model for PSC was developed that provides two-stage parcel sorting. A reduction in sorting time is achieved by minimising the number of ASC equipment settings that affect the redistribution of parcels from loading to unloading doors. In [13], two models were developed to address the problem of increasing conveyor throughput at a distribution centre while considering its operational constraints. The first model reduces sorting time by determining the minimum object transport route along the conveyor for a specific unloading–loading scenario. The second model implements a conveyor operating mode with maximum throughput and minimises all object transport routes for each unloading and loading operation. In [14], a deterministic mixed-integer linear model was developed for evaluating PSC operating scenarios. The modelling results were used to modify the conveyor configuration in order to reduce the time required for manual parcel transport. In [15], a model was developed for evaluating PSC conveyor throughput, enabling the assessment of various ASC layout options while considering parcel sorting time as a function of the number of loading and unloading doors.

To ensure the delivery of goods from manufacturers to consumers, a network of intermediate and final cross-docking distribution centers is used. Such distribution centers are designed for receiving trucks at inbound docks, unloading, sorting, and temporary storage of goods, followed by loading trucks at outbound docks for dispatch to destinations (the intermediate storage time for goods is limited to 24 hours). Goods are transported in standard pallets. ASCs (or ASLs) are used when it is necessary to unpack goods from delivered pallets for storage or to consolidate goods onto pallets for dispatch to a specified address. Research on cross-docking can be divided into several directions [2, 16–20]: studies on planning the location of distribution network nodes and goods delivery routing [2, 16]; studies on optimising the design and layout of cross-docking centre equipment [17, 18]; and studies on scheduling inbound and outbound trucks allocated to docks under conditions of limited dock availability and warehouse capacity [19, 20].

As a separate direction, one can identify studies [21–23] on the development of algorithms and decision-making models for ASLs that address the task of sorting objects with respect to their weight and dimensions. In [21], a nonlinear model for loading transport containers (sub-containers, pallets) was developed that solves the problem of object consolidation planning while considering their weight and geometric parameters (object orientation, dimensional characteristics of object placement within sub-containers). In [22], a fuzzy decision-making model for ASLs was developed, and in [23], an adaptive neuro-fuzzy inference system was developed, both enabling parcel sorting by weight and dimensions within a conveyor flow and implementing the specified loading logic for outbound trucks at the PSC.

This article is a continuation of the studies [22, 23].

3. Goal and objectives of the study

The goal of the study is to develop a decision-making model for the ASLs serving the loading doors of PSC terminals in the form of a neural network that performs flow-based classification of parcels into three ranges of weight and dimensions for implementing the specified sorting logic across loading doors and chutes.

To achieve the stated goals, the following tasks must be solved:

a) determine the weight and dimensional ranges of parcels associated with loading chute numbers;

b) determine the parcel sorting logic by weight and dimensions;

c) determine the parameters of datasets for training and testing neural networks;

d) conduct training and testing to determine the architecture of the following ANNs:

– Radial Basis Function Network (RBFNN);

– Generalized Regression Neural Network (GRNN);

– Probabilistic Neural Network (PNN);

– Feed-Forward Neural Network (FFNN);

e) conduct a comparative analysis of the neural networks using the number of neurons employed, the mean squared error (MSE) of training, and the relative precision of correct parcel classification by the weight and dimensional ranges defined by loading chute (loading door) numbers as quality metrics.

4. Materials and methods

Two PSC conveyor modification schemes [22, 23] are used in this study, enabling three variants of the specified parcel sorting logic for loading outbound trucks.

A simplified PSC conveyor scheme [22] implementing the first and second variants of outbound truck loading logic is presented in Fig. 1. Parcels from inbound trucks are unloaded at the PSC unloading doors designated as "U1", "U2", …, "UN". By means of sorting lines ASL-A and ASL-B of the first conveyor ASC-1, the primary sorting of parcels is performed across two terminals "A" or "B", each assigned to two subsequent delivery directions. For this purpose, the decision-making model of ASL-A (ASL-B) uses the delivery address information from the parcel label. To facilitate sorting at terminals "A" and "B", sorting lines ASL-1, ASL-2, …, ASL-N of conveyor ASC-A (ASC-B) are used, which are responsible for feeding parcels into the loading chutes "LC" of loading doors designated as "L1", "L2", …, "LN". Flow-based loading of parcels into outbound trucks is performed from chutes "LC" by manually stacking them in layers on top of one another, which entails a risk of parcel damage when they are placed on top of each other and results in inefficient utilisation of the vehicle cargo space.

To implement the specified loading logic according to the first and second variants, the decision-making models of sorting lines ASL-1, ASL-2, …, ASL-N of conveyors ASC-A and ASC-B must ensure parcel classification across N weight and dimensional ranges defined for each of the N loading doors. Figure 1 shows a variant for classifying parcels across 3 ranges. Such sorting provides two variants of loading logic:

– the first variant of loading logic is defined for a single outbound truck at terminal "A". For each loading door, ASL-1,2,3 selects parcels within the specified range. First, parcels whose parameters correspond to the maximum weight and dimensional range values (door "L3") are loaded into the truck. Then, the truck is sequentially loaded at doors "L2" and "L1", corresponding to decreasing weight and dimensional ranges. This ensures the possibility of compact truck cargo loading and reduces the risk of parcel damage when parcels are placed on top of one another;

– the second variant of loading logic is shown in Fig. 1 for several outbound trucks at terminal "B". In this variant, the simultaneous loading of 3 outbound trucks at doors "L1", "L2", …, "LN" with parcels of a single weight and dimensional parameter range is considered.

To implement the specified loading logic according to the third variant (Fig. 2), all loading doors "L" of terminals "A" and "B" are equipped with not one but several loading chutes "LC". Figure 2 shows a variant

ISSN 3083-7715 (online) Автоматизовані системи управління та прилади автоматики. 2026. № 1 (188)
ISSN 3083-7650 (0135-1710)(print) Management Information System and Devices. 2026. No. 1 (188)

with three chutes: "LC1", "LC2", "LC3". These chutes are designed to receive parcels in 3 weight and dimensional ranges. The range values increase in accordance with the order of loading chute "LC" numbers. In this case, the ASL decision-making model for loading doors does not differ from the first and second sorting logic variants (Fig. 1) and must ensure parcel classification across 3 weight and dimensional ranges defined for each of the 3 loading chutes "LC" (Fig. 2).

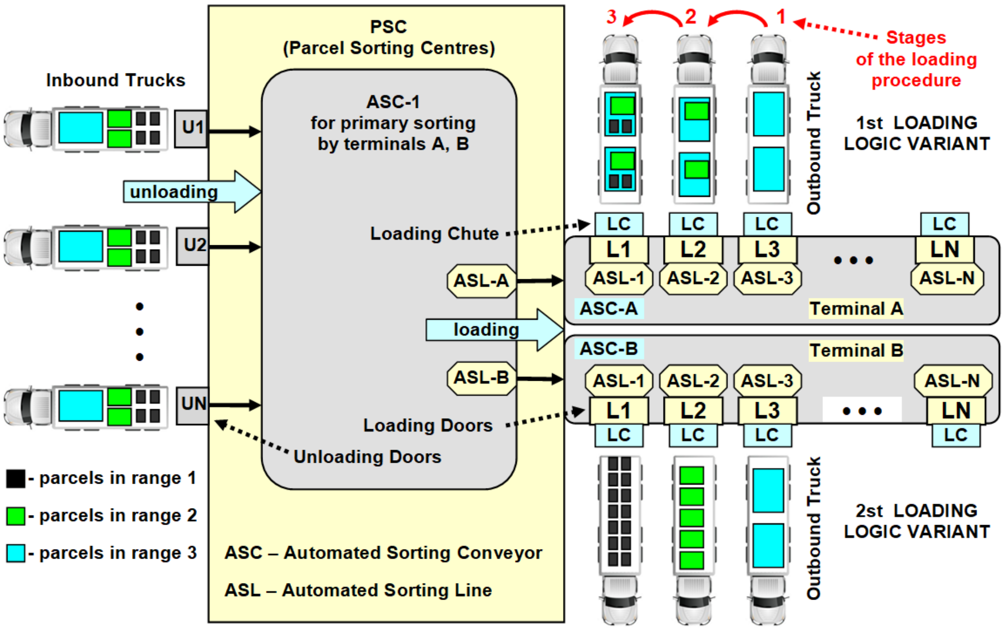

Fig. 1. First and second variants of truck loading logic

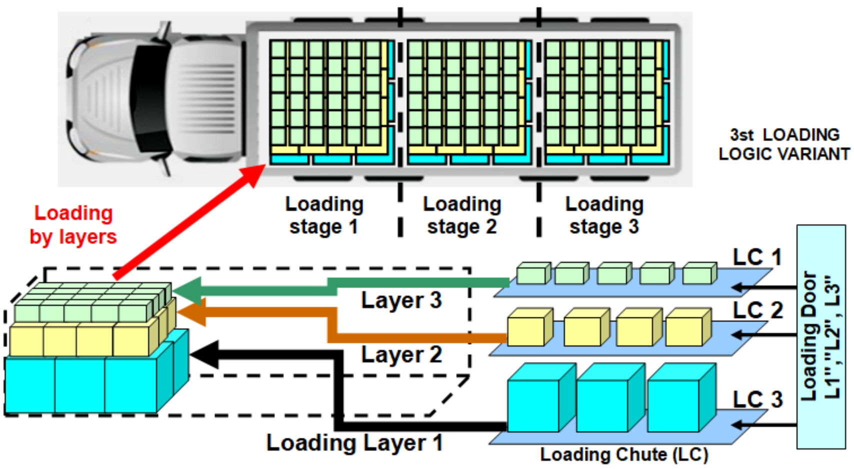

Fig. 2. Third variant of truck loading logic

Sorting across three loading chutes provides the third variant of parcel loading. Loading is performed in three or more stages depending on the volume of the vehicle cargo space. At each loading stage, parcels are arranged in three layers. The first layer consists of parcels with the greatest weight and dimensions from loading chute "LC3". The second layer consists of parcels in the medium range from chute "LC2", and the third layer consists of the lightest and smallest parcels from chute "LC1". In the second and subsequent loading stages, the process of arranging parcels by layers is repeated. This ensures the possibility of compact loading of the vehicle cargo space and reduces the risk of parcel damage when they are placed on top of one another.

Thus, in accordance with the goal and objectives of the study, it is necessary to develop a decision-making model for the ASLs serving the loading doors of PSC terminals in the form of a neural network that performs flow-based classification of parcels into three ranges of weight and dimensions for implementing the specified sorting logic across 3 loading doors or chutes.

The MATLAB environment toolkit (version R2016a) was employed for the development and investigation of neural network models using the computer simulation method. Enhanced training effectiveness of the neural networks under study was achieved through the use of a normalisation procedure for the input data of training and validation samples in accordance with the requirements of [24].

For the neural networks under investigation, the input data were specified as a set of column vectors of dimension $4 \times N$: $P_N = \left(p_{1n},\ p_{2n},\ p_{3n},\ p_{4n}\right)^T$, $n = 1,\ 2, \ldots, N$, where p_i denotes the parcel parameters: p_1 – weight, p_2 – height, p_3 – width, p_4 – depth. The target data (classification output) were defined as a scalar value representing the loading chute (door) number $Y_{Tn} \in \{1,\ 2,\ 3\}$, assigned to each vector P from the set P_N. The target data for the validation datasets Y_{Vn} were specified in an analogous manner. According to the problem formulation, the neural networks under investigation were required to perform classification of parcels into three loading chutes "LC" (or doors "L"). Each loading chute (door) number is associated with three value ranges for each of the four parcel parameters p_i. The assignment of one of the three ranges of each parameter of vector P to the loading chute or door numbers $Y_T\left(Y_V\right)$ is determined by the specified parcel sorting logic.

For the comparative analysis of the trained neural networks, the following quality assessment metrics were employed: the number of neurons utilized, the mean squared error of training $\left(MSE\right)$, and the relative precision $\left(RP\right)$ of correct classification of parcels into one of the three loading chute (door) numbers PSC. The RP was calculated during both the training $\left(RP_T\right)$ and validation $\left(RP_V\right)$ phases of the neural networks. For this purpose, the output values Y_{On} of the trained neural networks were rounded to integers in accordance with standard mathematical rules. Subsequently, the number of true classification $\left(TC\right)$ results was recorded during training $TC_T = count\left\{round\left(Y_{On}\right) = Y_{Tn}\right\}$ and validation $TC_V = count\left\{round\left(Y_{On}\right) = Y_{Vn}\right\}$. Analogously, the false classification $\left(FC\right)$ results were recorded during training $FC_T = count\left\{round\left(Y_{On}\right) \neq Y_{Tn}\right\}$ and validation $FC_V = count\left\{round\left(Y_{On}\right) \neq Y_{Vn}\right\}$, where the $count()$ function was used to calculate the number of satisfied conditions, and the $round()$ function was used to round a number to the nearest integer. The relative precision of correct classification was computed according to the following expression:

$$RP_{T(V)} = \frac{TC_{T(V)}}{TC_{T(V)} + FC_{T(V)}} = \frac{TC_{T(V)}}{N}, \tag{1}$$

where N is the number of input vectors in the training or validation datasets P_N.

5. Research Results and Discussion

5.1. Determination of Parcel Weight and Dimensional Ranges

According to the problem formulation, the neural networks under investigation were required to perform classification of parcels into three loading chutes "LC" (or doors "L"). Each loading chute (door) number is associated with three value ranges for each of the four parcel parameters p_i. The following parcel parameters are defined as neural network input data: p_1 – weight, p_2 – height, p_3 – width, and p_4 – depth. For each input parameter p_i, $i \in \{1,\ 2,\ 3,\ 4\}$, 4 value ranges are defined corresponding to 3 loading chutes "LC" (doors "L"). The parameter value ranges are designated as R_{1q}, R_{2j}, R_{3m}, R_{4s} where the first subscript is the parameter index p_i, and the second subscript $\left(q,\ j,\ m,\ s\right) \in \{1,\ 2,\ 3\}$ defines the range index, which coincides with the loading chute "LC" (door "L") number. The values of ranges $R_{i(q,j,m,s)}$ are given in Table 1.

In Table 1, the dimensional parameter ranges of parcels are chosen to be identical. This is done intentionally for the case when the QR code of a parcel label does not contain data on its dimensions. In this case, the ASC determines these parcel parameters from the results of processing their 3D images. However, during unloading, a parcel may be placed on the conveyor belt platform in an arbitrary orientation. Therefore, it is impossible to determine from the processing results which parameters $\{p_2, p_3, p_4\}$ the measured dimensional values correspond to.

ISSN 3083-7715 (online) Автоматизовані системи управління та прилади автоматики. 2026. № 1 (188)
ISSN 3083-7650 (0135-1710)(print) Management Information System and Devices. 2026. No. 1 (188)

Table 1. *Parcel parameter value ranges*

Input parameters	Range designation	Range of parcel parameter values for loading chutes (doors)		
		"LC1" ("L1")	"LC2" ("L3")	"LC3" ("L3")
p_1, weight (kg)	R_{1q}	$R_{11}-[0,\ 14.9]$	$R_{12}-[15,\ 44.9]$	$R_{13}-[45,\ 60]$
p_2, height (cm)	R_{2j}	$R_{21}-[0,\ 39.9]$	$R_{22}-[40,\ 119.9]$	$R_{23}-[120,\ 160]$
p_3, width (cm)	R_{3m}	$R_{31}-[0,\ 39.9]$	$R_{32}-[40,\ 119.9]$	$R_{33}-[120,\ 160]$
p_4, depth (cm)	R_{4s}	$R_{41}-[0,\ 39.9]$	$R_{42}-[40,\ 119.9]$	$R_{43}-[120,\ 160]$

5.2. Determination of parcel sorting logic

The parcel sorting logic is defined in accordance with the rules

$$\overset{K}{\underset{k=1}{\text{RULE}}}: IF\left(p_1 \in R_{1q}\right) \text{ and } IF\left(p_2 \in R_{2j}\right) \text{ and } IF\left(p_3 \in R_{3m}\right) \text{ and } IF\left(p_4 \in R_{4s}\right) \rightarrow L_k, \tag{2}$$

where $k=1,\ 2,\ \ldots,\ K$ are rule indices; $R_{i(q,j,m,s)}$ are the value ranges of parcel parameters p_i corresponding to three loading chutes "LC" (doors "L") with range numbers defined by the indices $(q,\ j,\ m,\ s)\in\{1,\ 2,\ 3\}$ (Table 1); $L_k=\{L_1,\ L_2,\ L_3\}=\{1,\ 2,\ 3\}$ is the loading chute "LC" (door "L") number.

The sorting logic was defined for each k-rule (2). For this purpose, four true conditions of belonging of a parcel with parameters $\{p_1, p_2, p_3, p_4\}$ to the corresponding ranges with indices $(q,\ j,\ m,\ s)\in\{1,\ 2,\ 3\}$ were considered, and the loading chute number L_k was assigned. The number of rules in (2) is determined by the expression: $K=Z^G=4^3=81$, where Z is the number of conditions in the rule, and G is the number of loading chutes. The sorting logic matrix is presented in Table 2 (in abridged form).

Table 2. *Parcel sorting logic matrix (abridged)*

Rule index k	Indexes of fulfilled conditions in rules (2)				Number "LC" ("L") for the k-th rule L_k
	$p_1 \in R_{1q}$	$p_2 \in R_{2j}$	$p_3 \in R_{3m}$	$p_4 \in R_{4s}$	
	index q	index j	index m	index s	
1	1	1	1	1	1
2	1	1	1	2	
3	1	1	1	3	
4	1	1	2	1	
5	1	1	3	1	
6	1	2	1	1	
7	1	3	1	1	
8	1	1	2	2	2
9	1	1	2	3	
10	1	1	3	2	
11	1	1	3	3	
12	1	2	1	2	
13	2	1	1	1	
...	...	...	...	...	
47	2	3	2	2	
48	2	1	3	3	3
49	2	2	3	3	
50	2	3	1	3	
51	2	3	2	3	
52	2	3	3	1	
53	3	1	1	1	
...	...	...	...	...	
81	3	3	3	3	

Автоматизовані системи управління та прилади автоматики. 2026. № 1 (188) *ISSN 3083-7715 (online)*
Management Information System and Devices. 2026. No. 1 (188) *ISSN 3083-7650 (0135-1710)(print)*

The sorting logic presented in Table 2 implements two criteria for verifying the truth of conditions for the belonging of parcel weight and dimensional values to the corresponding ranges associated with the three loading chutes (doors):

– first criterion: if the rule contains no more than one true condition with an index from $\{j,\ m,\ s\}$ whose value exceeds the index q, then the parcel is transported to the loading chute numbered q;

– second criterion: if the rule contains two or more indices of true conditions from $\{j,\ m,\ s\}$ that exceed the value of index q, then the parcel is transported to the loading chute numbered $q+1$. If $(q+1)>3$ then $q=3$.

5.3. Determination of dataset parameters for training and testing neural networks

For neural networks, the input data is a column vector P of dimension 4×1, and the target data Y_T is a scalar value representing the loading chute (door) number. For training the neural networks, 5 normalised datasets were used, which were formed as an input data array P of dimension $4\times N$. The vectors $P=(p_1, p_2, p_3, p_4)^T$ of array P_N contain the parcel parameter values, which were determined in accordance with Table 1. The normalisation of weight data p_1 and dimensional data p_2, p_3, p_4 was performed separately. The output (target) data is represented as a row vector $Y_T=\{L_{k1},\ L_{k2},\ ...,\ L_{kN}\}$ of loading chute (door) numbers, defined in accordance with Table 2 for each vector of array P_N. Validation of trained neural networks was performed using the "dataset6", which was formed in an analogous manner.

The parameters of the datasets for training and testing ANNs are presented in Table 3.

Table 3. *Parameters of datasets for training and testing ANNs*

Purpose of data sets	Number of values for ranges $R_{i(q,j,m,s)}$				Total number of tuples in the set (N)	Data dimension		Name of data sets
	R_{1q}	R_{2j}	R_{3m}	R_{4s}		$P_N\ (4\times N)$	$Y_T\ (1\times N)$	
Training	3	3	3	3	34=81	4×81	1×81	*dataset1*
	4	4	4	4	44=256	4×256	1×256	*dataset2*
	5	5	5	5	54=625	4×625	1×625	*dataset3*
	6	6	6	6	64=1296	4×1296	1×1296	*dataset4*
	9	9	9	9	94=6561	4×6561	1×6561	*dataset5*
Validation	13	13	13	13	134=28561	$P_N\ (4\times N)$	$Y_V\ (1\times N)$	*dataset6*
						4×28561	1×28561	

5.4. Investigation of Radial Basis Function (RBF) neural network models

The architecture of the RBF neural network model in MATLAB is presented in Fig. 3. The number of RBF network inputs is determined by the dimension R of the training data P_N. For the datasets P_N used (Table 3), with dimension $R\times N$, the RBF network has 4 inputs $(R=4)$.

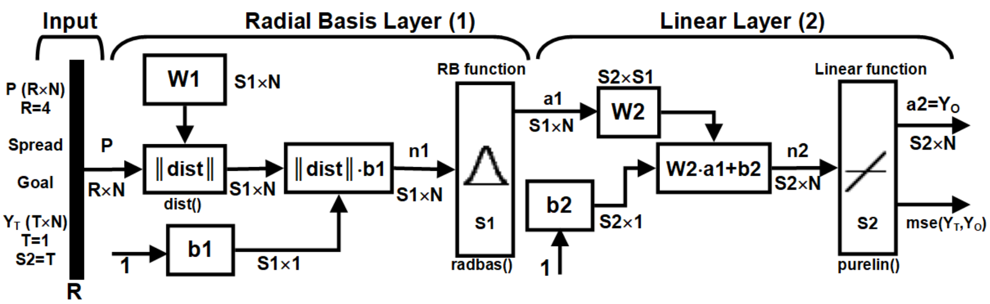

Fig. 3. Architecture of the RBF neural network model in MATLAB

ISSN 3083-7715 (online) Автоматизовані системи управління та прилади автоматики. 2026. № 1 (188)
ISSN 3083-7650 (0135-1710)(print) Management Information System and Devices. 2026. No. 1 (188)

The radial basis layer (Fig. 3) transforms the input data P using RB activation functions, centring their values with the weight vector $W1$ $(\text{net.IW}\{1,1\})$. The RB neurons compute the euclidean distance using the $dist()$ function, adding the bias b1 $(\text{net.b}\{1\})$:

$$\text{n1} = dist\left(\left(\text{net.IW}\{1,1\}, p\right), \text{net.b}\{1\}\right) = \|W1 - P\| \cdot \text{b1}. \quad (3)$$

The resulting RB neuron activation function has the form:

$$\text{a1} = \exp\left(-\left(\|W1 - P\| \cdot \text{b1}\right)^2 / spread\right) = radbas(\text{n1}), \quad (4)$$

where "Spread" is the smoothing parameter of the gaussian RB activation function that determines its width.

In the second layer, the output values a1 are linearly combined using weights $W2$ $(\text{net.LW}\{2,1\})$ and biases b2 $(\text{net.b}\{2\})$. The formed combinations $\text{n2} = W2 \cdot a1 + \text{b2}$, after activation by the linear function, form the RBF network output: $Y_O = \text{a2} = purelin(\text{n2})$. The number of linear layer neurons $S2$ (the number of RBF network outputs) is determined by the dimension T of the target dataset Y_T (for the problem under consideration, $S2 = T = 1$).

For training the RBF network, the input data P, target data Y_T, the minimum MSE value "Goal", and the "Spread" parameter are specified. The backpropagation method is used for training the RBF network. Training begins with one RB neuron $(S1 = 1)$. After determining the output value Y_O, the RBF network calculates the mean squared error between two vectors:

$$MSE = mse(Y_T, Y_O) = \frac{1}{N}\sum_{n=1}^{N}(Y_{Tn} - Y_{On})^2. \quad (5)$$

MSE minimisation is performed for each subsequent training epoch of the RBF network by iteratively adjusting the values of vectors $W1$, $W2$ and increasing the number of first-layer neurons $S1$ by "1" until the condition $MSE < Goal$ is satisfied.

Training and validation of RBF models for parcel classification research were performed using the datasets given in Table 3. The best training and testing results of RBF models, obtained for datasets "dataset1" and "dataset4", are presented in Table 4.

Table 4. *Results of RBF model training and validation*

No	Training							Validation			
	Data Set	$S1$	*Spread*	*Goal*	*MSE*	MSE_{TR}	RP_T	**Data Set**	MSE_V	MSE_{VR}	RP_V
1	*dataset1* (4×81)	74	0.1	0.01	0.0	0.0	1.0	*dataset6* (4×28561)	2.0283	2.1186	0.0418
2		67	0.3	0.01	0.0099	0.0	1.0		0.0609	0.0523	0.9477
3		43	0.5	0.01	0.0096	0.0	1.0		0.0739	0.0715	0.9285
4		50	0.7	0.01	0.0098	0.0	1.0		0.0765	0.0806	0.9194
5		40	0.9	0.01	0.0094	0.0	1.0		0.0925	0.1086	0.8914
6		53	1.0	0.01	0.0087	0.0	1.0		0.0944	0.1193	0.8807
7		45	1.5	0.01	0.0097	0.0	1.0		0.3693	0.4860	0.5604
8		45	2.0	0.01	0.0098	0.0	1.0		0.4535	0.5846	0.4850
9		63	0.3	0.02	0.0179	0.0	1.0		0.0639	0.0586	0.9414
10		58	0.3	0.03	0.0295	0.0617	0.9383		0.0672	0.0643	0.9357
11		55	0.3	0.04	0.0383	0.0370	0.9630		0.0695	0.0686	0.9314
12		51	0.3	0.05	0.0461	0.0494	0.9506		0.0696	0.0692	0.9308
13	*dataset4* (4×1296)	941	0.1	0.04	0.0400	0.0201	0.9799	*dataset6* (4×28561)	0.0813	0.0295	0.9705
14		882	0.1	0.05	0.0495	0.0918	0.9082		0.0797	0.0640	0.9360
15		240	0.3	0.01	0.0096	0.0	1.0		0.0683	0.0655	0.9345
16		137	0.3	0.03	0.0300	0.0046	0.9954		0.0725	0.0808	0.9192

In the course of the research, the values of relative precision of correct parcel classification RP (1) during training (RP_T) and validation (RP_V), the number of first-layer neurons $S1$, and four mean squared error values (Table 4) were recorded: MSE (5), MSE_{TR}, MSE_V, MSE_{VR}. The MSE_{TR} values were calculated using the formula:

$$MSE_{TR} = mse(Y_T,\ Y_O) = \frac{1}{N}\sum_{n=1}^{N}\left(round\ (Y_{Tn}) - round\ (Y_{On})\right)^2, \quad (6)$$

where the round() function defines the rounding operation to the nearest integer according to mathematical rules.

The values $MSE_V = mse(Y_V, Y_O)$ and $MSE_{VR} = mse(Y_V, Y_O)$ were calculated for the validation output Y_O and target Y_V data using (5) and (6).

The use of MSE_V and MSE_{VR} is associated with the fact that the relative precision of parcel classification RP_T, RP_V by loading chute (door) numbers is determined from the RBF network output data after rounding them to integers; therefore, the standard MSE values are uninformative. As the study showed (rows 1–10, 15 of Table 4), achieving zero standard MSE values does not lead to expected results, which is explained by the presence of errors exceeding ±0,5. The value of the specified error "Goal" determines the number of first-layer neurons: as it increases, the $S1$ value decreases (rows 2, 9–12 of Table 4). When using large-size samples and decreasing the "Goal" value (rows 13–16 of Table 4), the number of neurons $S1$ can reach the dimension N of the training sample.

In accordance with the problem statement, the RBF model (row 2 of Table 4) was selected, providing the required relative precision of correct parcel classification $\mathrm{RP_V} = 95\,\%$ with a minimum number of neurons $\mathrm{S1} = 67$.

5.5. Investigation of Generalized Regression Neural Network (GRNN) models

The architecture of the GRNN model in MATLAB is presented in Fig. 4. Its first layer is analogous to the RBF network architecture (Fig. 3) but differs in the principle of forming the weight vector $W1$ $(\mathrm{net.IW}\{1,1\})$ values. These values exactly coincide with the training sample P_N data values. Each neuron in the first layer computes the Euclidean distance and the radial basis function in accordance with (3) and (4).

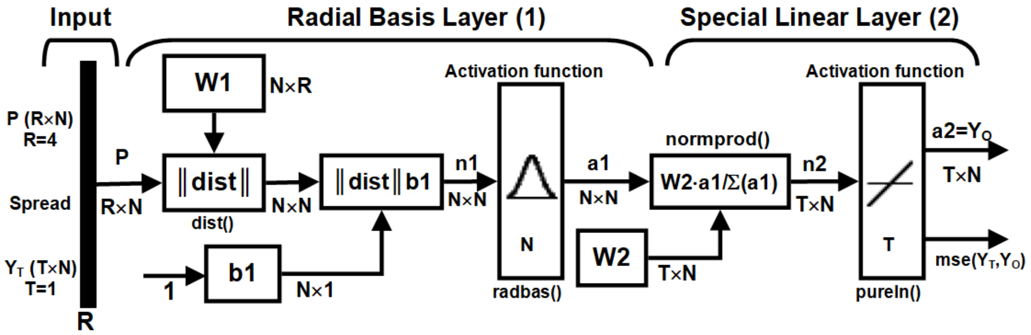

Fig. 4. Architecture of the GRNN model in MATLAB

The GRNN linear neuron layer computes the weighted mean of the outputs a1 associated with each input vector P and the weights of vector $W2$ $(\mathrm{net.LW}\{2,1\})$. For this purpose, the normalised dot product is calculated:

$$\mathrm{n2} = \frac{W2 \cdot \mathrm{a1}}{\sum \mathrm{a1}} = \frac{\mathrm{net.LW}\{2,1\} \cdot \mathrm{a1}}{sum(\mathrm{a1})} = normprod(\mathrm{net.LW}\{2,1\}, \mathrm{a1}).$$

The formed weighted outputs n2, after activation by the linear function, form the GRNN output: $Y_O = \mathrm{a2} = purelin(\mathrm{n2})$. The number of GRNN linear layer neurons T is determined by the dimension of the target dataset Y_T (for the problem under consideration, $T = 1$).

Training and validation of GRNN models for parcel classification research were performed using the datasets given in Table 3. GRNN training by the backpropagation method does not include interactive adjustment of the number of first-layer neurons, which is determined by the dimension N of the input training data P_N and weights $W1$. The best training and testing results of GRNN models, obtained for datasets "dataset1", "dataset2", "dataset3", "dataset4" are presented in Table 5.

ISSN 3083-7715 (online) Автоматизовані системи управління та прилади автоматики. 2026. № 1 (188)
ISSN 3083-7650 (0135-1710)(print) Management Information System and Devices. 2026. No. 1 (188)

Table 5. *Results of GRNN model training and validation*

No	Training						Validation			
	Data Set	$N1$	*Spread*	MSE	MSE_{TR}	RP_T	**Data Set**	MSE_V	MSE_{VR}	RP_V
1	*dataset1* (4×81)	81	0.1	0.0	0.0	1.0	*dataset6* (4×28561)	0.0228	0.0126	0.9874
2		81	0.2	0.0	0.0	1.0		0.0459	0.0329	0.9671
3		81	0.3	0.0261	0.0	1.0		0.0831	0.0738	0.9262
4		81	0.5	0.1191	0.1235	0.8765		0.1375	0.1531	0.8469
5		81	0.7	0.1983	0.2222	0.7778		0.1839	0.2576	0.7424
6	*dataset1* (4×81)	81	0.9	0.2562	0.3827	0.6173	*dataset6* (4×28561)	0.2182	0.3981	0.6019
7		81	1.0	0.2773	0.4568	0.5432		0.2307	0.4333	0.5667
8	*dataset2* (4×256)	256	0.1	0.0001	0.0	1.0	*dataset6* (4×28561)	0.2358	0.2682	0.7318
9		256	0.3	0.0640	0.0234	0.9766		0.1350	0.2147	0.7853
10		256	0.5	0.1935	0.3594	0.6406		0.1625	0.2447	0.7553
11		256	1.0	0.4490	0.6250	0.3750		0.2816	0.4431	0.5569
12	*dataset3* (4×625)	625	0.1	0.1052	0.1744	0.8256	*dataset6* (4×28561)	0.1618	0.1989	0.8011
13		625	0.2	0.1291	0.1808	0.8192		0.1219	0.1755	0.8245
14		625	0.3	0.1553	0.1952	0.8048		0.1163	0.1620	0.8380
15		625	0.5	0.2179	0.3456	0.6544		0.1678	0.2951	0.7049
16	*dataset4* (4×1296)	1296	0.1	0.0030	0.0	1.0	*dataset6* (4×28561)	0.0485	0.0629	0.9371
17		1296	0.3	0.0957	0.1057	0.8943		0.0965	0.1013	0.8987
18		1296	1.0	0.3158	0.5054	0.4946		0.2423	0.4429	0.5571

During the GRNN model research (similarly to Table 4), the values of relative precision of correct parcel classification RP_T, RP_V, the "Spread" parameter value, the number of first-layer neurons $N1$, and four mean squared error values MSE, MSE_{TR}, MSE_V, MSE_{VR} were recorded.

As the study showed, for each training sample the "Spread" parameter must be selected to achieve the maximum value of relative parcel parameter classification precision RP_V by loading chute numbers (ranges). For training datasets "dataset1", "dataset4" (rows 1–7, 16–18 of Table 5), the RP_V value decreases with increasing "Spread" value, and the maximum RP_V value was obtained for $Spread = 0{,}1$. For training datasets "dataset2", "dataset3" (rows 8–15 of Table 5), the maximum RP_V value was obtained for $Spread = 0{,}2$. The RP_V values for training datasets "dataset2", "dataset3", "dataset4" are lower than for "dataset1", despite their larger dimension. A deterioration in validation results was also observed (rows 1–2 of Table 5) when zero MSE values were achieved during training, which was caused by the presence of errors exceeding ±0,5.

In accordance with the problem statement, the GRNN model (row 1 of Table 5) was selected, providing the required relative precision of correct parcel classification $RP_V = 99\%$ with a minimum number of neurons $N1 = 81$.

5.6. Investigation of probabilistic neural network (PNN) models

The architecture of the PNN model in MATLAB is presented in Fig. 5. Training data, containing N vectors P for each of which the target data Y_T defines numerical class values in the form of loading chute (door) numbers $L_k = \{1,\ 2,\ 3\}$, is fed to the PNN input. To facilitate classification, the function $ind2vec(Y_T)$ converts the dimensionality of the target data Y_T from the $1 \times N$ format into a positional target vector matrix T of dimension $G \times N$. The parameter $G_{\max} = 3$ specifies the number of classifiable vectors and is determined by the maximum value of the loading chute (door) number L_k, $k = \{1,\ 2,\ 3\}$. The $ind2vec()$ function converts the numerical class values $\{1,\ 2,\ 3\}$ into three vectors $\left\{(1\ 0\ 0)^{\mathrm{T}},\ (0\ 1\ 0)^{\mathrm{T}},\ (0\ 0\ 1)^{\mathrm{T}}\right\}$, respectively.

The first PNN layer (Fig. 5) is analogous to the GRNN architecture (Fig. 4), where the weight vector $W1$ $(\text{net.IW}\{1,1\})$ values exactly coincide with the training sample P_N values. Each neuron in the first layer computes the Euclidean distance and the radial basis function in accordance with (3) and (4).

The second-layer weight matrix $W2$ $(\text{net.LW}\{2,1\})$ corresponds to the target vector matrix T. The product $W2 \cdot \text{a1}$ determines N elements of matrix n2 and their numerical correspondence to each of the G classes. The competing activation function $compet(\text{n2})$ creates the output data array a2. This array contains only one value equal to "1" for the largest element value of vector n2 and "0" in all other cases. The $vec2ind(\text{a2})$ function is used to convert the positional matrix a2 to the numerical class representation.

Training and validation of PNN models for parcel classification research were performed using the datasets given in Table 3. The best training and testing results of PNN models, obtained for datasets "dataset1"–"dataset4", are presented in Table 6.

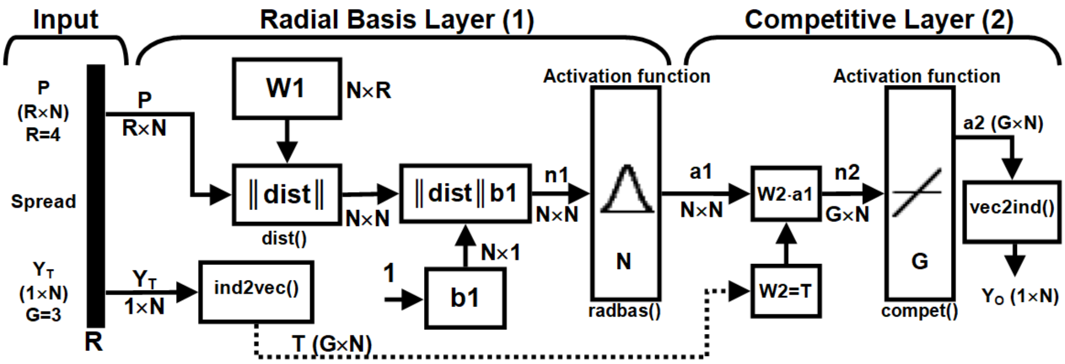

Fig. 5. Architecture of the PNN model in MATLAB

Table 6. *Results of PNN model training and validation*

No	Training					Validation		
	Data Set	$N1$	*Spread*	MSE_{TR}	RP_T	Data Set	MSE_{VR}	RP_V
1	*dataset1* (4×81)	81	0,1	0.00	1.0	*dataset6* (4×28561)	0.0126	0.9874
2		81	0,2	0.00	1.0		0.0329	0.9671
3		81	0,3	0.00	1.0		0.0736	0.9264
4		81	0,5	0.1235	0.8765		0.1303	0.8697
5		81	0,7	0.1605	0.8395		0.1485	0.8515
6		81	0,9	0.1728	0.8272		0.1903	0.8097
7		81	1,0	0.1728	0.8272		0.2203	0.7797
8	*dataset2* (4×256)	256	0,1	0.00	1.0	*dataset6* (4×28561)	0.2764	0.7236
9		256	0,3	0.00	1.0		0.2712	0.7288
10		256	0,5	0.1484	0.8516		0.2524	0.7476
11		256	1,0	0.3555	0.6445		0.2164	0.7836
12	*dataset3* (4×625)	625	0,1	0.1744	0.8256	*dataset6* (4×28561)	0.1996	0.8004
13		625	0,2	0.1808	0.8192		0.1838	0.8162
14		625	0,5	0.1904	0.8096		0.1612	0.8388
15		625	1,0	0.4704	0.5296		0.4398	0.5602
16	*dataset4* (4×1296)	1296	0,1	0.00	1.0	*dataset6* (4×28561)	0.0629	0.9371
17		1296	0,3	0.1057	0.8943		0.1035	0.8965
18		1296	1,0	0.2917	0.7083		0.2411	0.7589

During the PNN model research, the values of relative precision of correct parcel classification during training (RP_T) and validation (RP_V), the "Spread" parameter value, and the number of first-layer neurons $N1$ were recorded. Unlike the previously considered neural networks, the model output data are integers and do not require rounding. Therefore, only MSE_{TR} and MSE_{VR} were calculated.

For training datasets "dataset1", "dataset4" (rows 1–7, 16–18 of Table 6), the RP_V value decreases with increasing "Spread" value, and the maximum RP_V value was obtained for $Spread = 0{,}1$. The maximum RP_V value for training dataset "dataset2" (rows 8–11 of Table 6) was obtained for $Spread = 1.0$, and for "dataset3" (rows 12–15 of Table 6), $Spread = 0{,}5$.

ISSN 3083-7715 (online) Автоматизовані системи управління та прилади автоматики. 2026. № 1 (188)
ISSN 3083-7650 (0135-1710)(print) Management Information System and Devices. 2026. No. 1 (188)

The RP_V values for training datasets "dataset2", "dataset3", "dataset4" are lower than for "dataset1", despite their larger dimension. The minor differences in the obtained data between the PNN (Table 6) and GRNN (Table 5) models are explained by the fact that the architectures of their first RB layer are completely analogous.

In accordance with the problem statement, the PNN model (row 1 of Table 6) was selected, providing the required relative precision of correct parcel classification $RP_V = 99\%$ with a minimum number of neurons $N1 = 81$.

5.7. Investigation of Feed-Forward Neural Network (FFNN) Models

The architecture of the FFNN model in MATLAB is presented in Fig. 6.

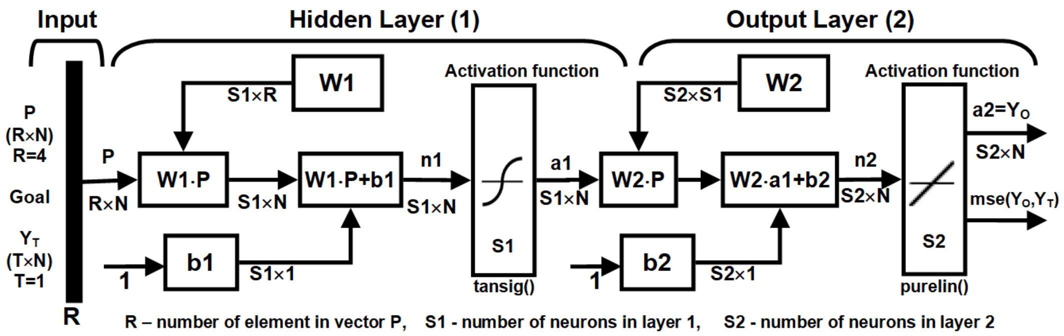

Fig. 6. Architecture of the FFNN model in MATLAB

For the datasets P_N used (Table 3), with dimension $R \times N$, the FFNN has 4 inputs $(\mathrm{R}=4)$. In the first (hidden) layer, the P values are linearly combined using weights $W1$ $(\mathrm{net.IW}\{1,1\})$ and biases $\mathrm{b1}$ $(\mathrm{net.b}\{1\})$. The formed combinations $\mathrm{n1} = W1 \cdot P + \mathrm{b1}$, after activation by the hyperbolic tangent function, form the hidden layer output of the FFNN: $\mathrm{a1} = tansig(\mathrm{n1})$. In the second layer, the values $\mathrm{a1}$ are linearly combined using weights $W2$ $(\mathrm{net.LW}\{2,1\})$ and biases $\mathrm{b2}$ $(\mathrm{net.b}\{2\})$. The combinations $\mathrm{n2} = W2 \cdot \mathrm{a1} + \mathrm{b2}$, after activation by the linear function, form the FFNN output: $Y_O = \mathrm{a2} = purelin(\mathrm{n2})$. The number of linear layer neurons $S2$ (the number of FFNN outputs) is determined by the dimension T of the target dataset Y_T (for the problem under consideration, $S2 = T = 1$).

The backpropagation method is used for training the FFNN. During FFNN training, the weights $W1$, $W2$ (Fig. 6) of the hidden and output FFNN layers are adjusted. The MATLAB toolkit implements 14 training algorithms. Based on the research results, two of them were selected: the Levenberg–Marquardt (LM) algorithm and the Bayesian Regularisation (BR) algorithm.

For conducting parcel classification research using the FFNN, the following were specified: the training algorithm (LM, BR), the number of first-layer neurons $S1$, input data P_N, and target data Y_T (Table 3). Table 7 presents a fragment of the FFNN model training and testing results.

During the FFNN model research, the values of relative precision of correct parcel classification RP_T, RP_V, the number of first-layer neurons $S1$, and four mean squared error values MSE, MSE_{TR}, MSE_V, MSE_{VR} were recorded. As the FFNN modelling demonstrated, for the two training algorithms used and training datasets with dimension N less than 1296, unsatisfactory values at the level of 90% or below were obtained (rows 1–4, 10–12 of Table 7). Increasing the volume of training datasets yields only a slight MSE decrease, with errors still exceeding ±0.5, and an RP_V gain of only 0.1–0.3%.

In accordance with the problem statement, the FFNN model (row 9 of Table 7) trained using the Levenberg–Marquardt algorithm was selected, providing the required relative precision of correct parcel classification $RP_V = 97\%$ with a minimum number of first-layer neurons $S1 = 81$.

Table 7. *Results of FFNN model training and validation*

No	Training					Validation			
	Data Set	$S1$	MSE	MSE_{TR}	RP_T	**Data Set**	MSE_V	MSE_{VR}	RP_V
Levenberg–Marquardt (LM) training method									
1	*dataset1* (4×81)	20	0.0416	0.0617	0.9383	*dataset6* (4×28561)	0.1118	0.1381	0.8628
2		24	0.0357	0.0247	0.9753		0.1125	0.1517	0.8483
3		28	0.1004	0.1111	0.8889		0.1542	0.1804	0.8196
4	*dataset4* (4×1296)	20	0.0305	0.0193	0.9807	*dataset6* (4×28561)	0.0628	0.0692	0.9308
5		24	0.0327	0.0270	0.9730		0.0636	0.0710	0.9290
6		28	0.0484	0.0255	0.9745		0.0802	0.1061	0.8939
7	*dataset5* (4×6561)	20	0.0449	0.0453	0.9547	*dataset6* (4×28561)	0.0540	0.0550	0.9450
8		24	0.0510	0.0590	0.9410		0.0448	0.0396	0.9604
9		28	0.0447	0.0430	0.9570		0.0403	0.0302	0.9698
Bayesian Regularization (BR) training method									
10	*dataset1* (4×81)	20	0.0406	0.0370	0.9630	*dataset6* (4×28561)	0.1010	0.1106	0.8894
11		24	0.0422	0.0741	0.9259		0.0990	0.0908	0.9092
12		28	0.0368	0.0494	0.9506		0.0985	0.1082	0.8918
13	*dataset4* (4×1296)	20	0.0257	0.0039	0.9961	*dataset6* (4×28561)	0.0703	0.0668	0.9332
14		24	0.0198	0.0039	0.9961		0.0694	0.0696	0.9304
15		28	0.0171	0.0031	0.9969		0.0754	0.0731	0.9269
16	*dataset5* (4×6561)	20	0.0369	0.0346	0.9654	*dataset6* (4×28561)	0.0680	0.0647	0.9353
17		24	0.0354	0.0373	0.9627		0.0729	0.0828	0.9172
18		28	0.0344	0.0340	0.9660		0.1732	0.1883	0.8437

5.8. Discussion of Research Results

In accordance with the problem statement, an investigation of neural network models was conducted, using as quality metrics the minimum number of first-layer neurons $N1$, MSE, and relative precision of correct parcel classification RP_V by the weight and dimensional ranges defined by loading chute (loading door) numbers. The neural networks selected based on the research results are presented in Table 8.

An analysis of the obtained results shows that the GRNN and PNN networks do not satisfy the selection criteria. Their principal disadvantage is that during training, the number of neurons in their hidden (first) layer is determined by the number of elements in the training dataset.

Table 8. *Comparative evaluation results of trained neural network models*

Вид NN	$N1$	MSE	MSE_{TR}	RP_T	MSE_V	MSE_{VR}	RP_V
RBF	67	0.0099	0.0	1.0	0.0609	0.0523	0.9477
GRNN	81	0.0	0.0	1.0	0.0228	0.0126	0.9874
PNN	81	–	0.00	1.0	1.0	0.0126	0.9874
FFNN	28	0.0447	0.0430	0.9570	0.0403	0.0302	0.9698

It should be noted that the implementation of exponential radial functions for the GRNN, PNN, and RBF networks leads to a slowdown in their operation due to large computational volumes. In this regard, the FFNN is the most promising, as it provides the required relative precision of correct parcel classification $(RP_V = 97\%)$, and its hidden layer is implemented using the minimum number of neurons $(N1 = 28)$ compared to the other neural networks.

The computational volume of the trained FFNN is associated with vector addition and multiplication operations, which significantly simplifies its software implementation. The values of the weight matrix $W1$ (28×4) and weight vectors $W2$ (28×1), b1 (28×1), and b2 for the software implementation of the trained FFNN are presented in Table 9 (in the table, the matrix and vector elements are presented in transposed form).

ISSN 3083-7715 (online) Автоматизовані системи управління та прилади автоматики. 2026. № 1 (188)
ISSN 3083-7650 (0135-1710)(print) Management Information System and Devices. 2026. No. 1 (188)

Table 9. *Weight coefficient values of the trained FFNN model*

$W1^T\,(4\times 28)$																											
1,6	-0,4	-0,5	-0,1	-5,0	-3,2	-9,2	-14,9	-6,5	-0,6	-13,1	2,7	1,2	-0,8	-1,8	-0,9	11,5	-1,2	-3,7	-1,2	-1,3	-0,5	0,3	-21,1	-1,5	5,0	1,3	1,5
0,0	-0,2	-0,6	0,6	-1,0	0,0	-0,7	-0,6	7,2	0,4	0,3	-0,6	-0,4	1,9	-2,2	2,0	-0,5	-0,7	0,2	6,8	-0,5	0,3	0,5	-1,6	0,8	-1,0	-0,7	0,7
0,3	1,2	1,6	1,4	-1,1	3,4	-0,8	-0,6	0,1	1,0	0,2	-0,9	0,9	0,0	0,1	0,1	-0,5	-0,1	-4,3	-2,0	-0,2	-1,1	-0,8	-1,7	0,6	-0,6	-0,6	1,2
7,6	-0,1	-0,7	-0,7	-0,8	-0,3	-0,6	-0,5	0,0	-0,4	0,2	-1,6	-1,4	-1,2	0,2	-1,4	-0,4	1,2	-4,5	0,6	1,3	-0,6	-0,2	-1,6	0,0	1,8	0,2	1,5
$b1^T\,(1\times 28)$																											
-4,3	1,3	-1,8	2,3	1,0	1,6	3,5	6,6	3,1	0,6	-6,7	-0,7	0,4	-0,3	-0,9	-0,4	5,9	-0,5	-1,8	3,7	-0,5	-0,7	-1,1	-10,7	-2,2	5,3	1,9	2,9
$W2^T\,(1\times 28)$																											
0,1	-1,7	-0,2	0,8	-0,5	-0,1	2,5	-2,6	-0,1	0,6	-2,4	0,2	0,3	0,8	0,3	-0,6	-1,9	-1,8	-0,1	0,0	2,2	0,4	-1,8	0,3	-2,7	0,2	-3,2	0,5
$b2 = -0.59$																											

6. Conclusions and prospects for further research

The conducted study made it possible to determine the advantages and limitations of the neural network architectures selected for implementing the ASL decision-making model that classifies parcels into three ranges of weight and dimensions and implements the specified logic of their sorting across PSC loading doors and chutes within a conveyor flow.

A comparative analysis of such two-layer neural network architectures as RBF, GRNN, and PNN demonstrated that the FFNN provides the required relative precision of correct parcel classification within a conveyor flow according to their weight and dimensional ranges defined for PSC loading doors and chutes, with the minimum number of neurons employed in the hidden layer.

It is particularly noteworthy that the GRNN and PNN, while capable of achieving higher levels of relative precision of correct parcel classification, are limited in their applicability by the number of neurons in their hidden (first) layer, which is determined by the number of elements in the training dataset. A further disadvantage of the GRNN, PNN, and RBF is the significant computational volume caused by the large number of neurons and the use of exponential radial functions; however, such neural networks may prove useful in other application scenarios.

Considering the identified shortcomings, the following directions for further research can be defined:

– conducting research into possible scenarios for using decision-making models based on GRNN, PNN, and RBF to implement parcel sorting planning prior to their arrival at the PSC. With a prepared sorting plan, it is possible to implement short parcel transport routes along the PSC conveyor for their loading with consideration of weight and dimensional parameters;

– conducting research into possible scenarios for using FFNN, GRNN, PNN, and RBF at cross-docking transhipment distribution centers to implement models for planning preliminary goods sorting, as well as ASL decision-making models for direct goods sorting. Such use of neural networks enables the planning of goods unloading and their distribution across storage racks. An alternative scenario involves planning goods sorting using their labels, where the ASL is used for consolidating goods onto pallets with consideration of weight and dimensions.

Thus, the comparative analysis not only identified the most effective neural network architectures but also revealed their shortcomings and limitations. The research results demonstrated that solving the problem of parcel sorting within a conveyor flow requires implementing an ASL decision-making model based on the FFNN, which will leverage the advantages of its architecture.

Conflict of interest

The authors declare that they have no conflict of interest with respect to this research, including financial, personal, authorship, or other conflicts that could influence the research and its results presented in this article.

Funding

The study was conducted without financial support.

Data availability

The manuscript has no associated data.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technology in the creation of this work.

References

1. McWilliams, D., Stanfield, P, Geiger, C. (2005), "The Parcel Hub Scheduling Problem: A Simulation-Based Solution Approach", *Computers & Industrial Engineering*, Vol. 49(3). P. 393–412. DOI: https://doi.org/10.1016/j.cie.2005.07.002

2. Akkerman, F., Lalla-Ruiz, E, Mes, M., Spitters, T. (2022), "Cross-Docking: Current Research Versus Industry Practice and Industry 4.0 Adoption", *Smart Industry – Better Management*, Vol. 28, P. 70–104. DOI: https://doi.org/10.1108/S1877-636120220000028007

3. Soni, B., Mathur, P., Bora, A. (2021), "In Depth Analysis, Applications and Future Issues of Artificial Neural Network", *In: Enabling AI Applications in Data Science. Studies in Computational Intelligence*, Vol. 911, P. 149–183. Springer, Cham. DOI:https://doi.org/10.1007/978-3-030-52067-0_7

4. Katal, A., Singh, N. (2022), "Artificial Neural Network: Models, Applications, and Challenges", *In: Innovative Trends in Computational Intelligence. EAI/Springer Innovations in Communication and Computing*. P. 235–257. Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-78284-9_11

5. Daniel, G. (2019), "Principles of Artificial Neural Networks. Basic Designs to Deep Learning", *The World Scientific Press*, 440 p. ISBN: 978-981-12-0122-6. DOI: https://doi.org/10.1142/11306

6. Bugow, S., Kellenbrink, C. (2023), "The Parcel Hub Scheduling Problem With Limited Conveyor Capacity and Controllable Unloading Speeds", *OR Spectrum*, No 45, P. 325–357. DOI: https://doi.org/10.1007/s00291-022-00702-y

7. McWilliams, D., Stanfield, P., Geiger, C. (2008), "Minimizing the completion time of the transfer operations in a central parcel consolidation terminal with unequal-batch-size inbound trailers", *Computers & Industrial Engineering*, Vol. 54 (4), P. 709–720, https://doi.org/10.1016/j.cie.2007.10.006

8. Chen J., Chen, T., Ou, T., Lee, Y. (2019), "Adaptive Genetic Algorithm for Parcel Hub Scheduling Problem With Shortcuts in Closed-Loop Sortation System". *Computers & Industrial Engineering*. Vol. 138:106114. DOI: https://doi.org/10.1016/j.cie.2019.106114

9. Chen, J., Chen, T., Lee, Y. (2023), "Simulation Optimization For Parcel Hub Scheduling Problem in Closed-Loop Sortation System with Shortcuts", *Simulation Modelling Practice and Theory*, Vol. 124(10):102728. DOI: https://doi.org/10.1016/j.simpat.2023.102728

10. Grebennik, I., Dupas, R., Lytvynenko, O., Urniaieva, I. (2017), "Scheduling Freight Trains in Rail-rail Transshipment Yards with Train Arrangements", *International Journal of Intelligent Systems and Applications (IJISA)*, Vol. 9(10), P. 12–19. DOI: https://doi.org/10.5815/ijisa.2017.10.02

11. Chen, T., Chen, J., Huang, C., Chang, P. (2021), "Solving the Layout Design Problem by Simulation-Optimization Approach – A Case Study on a Sortation Conveyor System", *Simulation Modelling Practice and Theory*, Vol. 106:102192. DOI: https://doi.org/10.1016/j.simpat.2020.102192

12. Jarrah, A., Xiangtong, Q., Bard, J. (2014), "The Destination-Loader-Door Assignment Problem for Automated Package Sorting Centers", *Transportation Science*, No 50(4), P. 1314–1336. DOI: https://doi.org/10.1287/trsc. 2014.0521

13. Karami, F., Fathi, M., Pardalos, P. (2023), "Conveyor Operations in Distribution Centers: Modeling and Optimization", *Optim Lett* , Vol. 17, P. 1049–1068. DOI: https://doi.org/10.1007/s11590-022-01912-7

14. Werners, B., Wülfing, T. (2010), "Robust Optimization of Internal Transports at a Parcel Sorting Center Operated by Deutsche Post World Net", *Eur. J. Oper. Res* , Vol. 201(2), P. 419–426. DOI: https://doi.org/10.1016/j.ejor.2009.02.035

15. Fedtke, S. , Boysen, N. (2014), Layout Planning of Sortation Conveyors in Parcel Distribution Centers. *Transportation Science*, Vol. 51(1), P: 3–18. DOI: https://doi.org/10.1287/trsc.2014.0540

16. Lee, Y., Jung, J., Lee, K. (2006), Vehicle Routing scheduling for Cross-Docking in the Supply Chain, *Computers & Industrial Engineering*, Vol. 51(2), P. 247–256. DOI: https://doi.org/10.1016/j.cie.2006.02.006

17. McAree, P., Bodin, L., Ball, M. (2002), "Models for the Design and Analysis of a Large Package Sort Facility. Networks", Vol. 39, P. 107–120. DOI: https://doi.org/10.1002/net.10017

18. Bozer, Y., Hsieh, Y. (2004), "Expected Waiting Times at Loading Stations in Discrete-space Closed-loop Conveyors", *European Journal of Operational Research*, Vol. 155(2), P. 516–532. DOI: https://doi.org/10.1016/S0377-2217(02)00886-X

19. Oh, Y., Cha, C., Lee, S. (2006), "A Dock-door Assignment Problem for the Korean Mail Distribution Center", *Computers & Industrial Engineering*, Vol. 51(2), P. 288–296. DOI: https://doi.org/10.1016/j.cie.2006.02.009

20. Choy, K., Chow, H., Poon, T., Ho, G. (2012), "Cross-dock Job Assignment Problem in Space-constrained Industrial Logistics Distribution Hubs with a Single Docking Zone", *International Journal of Production Research*, Vol. 50(9), P. 2439–2450. DOI: https://doi.org/10.1080/00207543.2011.581006

21. Romanova, T., Litvinchev, I., Grebennik, I., Kovalenko, A., Urniaieva, I. (2020), "Packing Convex 3D Objects with Special Geometric and Balancing Conditions", *Advances in Intelligent Systems and Computing*, Vol. 1072. DOI: https://doi.org/10.1007/978-3-030-33585-4_27

22. Grebennik, I., Kovalenko, O. (2024), "Realisation of a Given Trucks Loading Logic using a Fuzzy Decision Making Model", *14th International Conference on Advanced Computer Information Technologies (ACIT)*, IEEE, P. 27–31. DOI: https://doi.org/10.1109/ACIT62333.2024.10712615

23. Grebennik, I., Kovalenko, O. (2025), "Adaptive Neuro-Fuzzy Inference System (ANFIS) for Control of Loading Logic in Parcel Sorting Centers", *15th International Conference on Advanced Computer Information Technologies (ACIT)*, IEEE, P. 46–50, DOI: https://doi.org/10.1109/ACIT65614.2025.11185826

24. Nawi, N., Atomi, W., Rehman, M. (2013), "The Effect of Data Pre-Processing on Optimized Training of Artificial Neural Networks, Procedia Technology", Vol. 11, P. 32–39. DOI: https://doi.org/10.1016/j.protcy.2013.12.159

Received (Надійшла) 09.02.2026
Accepted for publication (Прийнята до друку) 20.02.2026
Publication date (Дата публікації) 12.03.2026

Відомості про авторів / About the Authors

Гребеннік Ігор Валерійович – доктор технічних наук, професор, Харківський національний університет радіоелектроніки, завідувач кафедри комп'ютерного моделювання та інтелектуальних технологій; Харків, Україна;
Igor Grebennik – Doctor of Technical Sciences, Professor, Kharkiv National University of Radio Electronics, Head at the Computer Modelling and Intelligent Technologies Department; Kharkiv, Ukraine;
e-mail: igor.grebennik@nure.ua
ORCID ID: https://orcid.org/0000-0003-3716-9638

Коваленко Олексій Андрійович – Харківський національний університет радіоелектроніки, аспірант кафедри комп'ютерного моделювання та інтелектуальних технологій; Харків, Україна;
Oleksii Kovalenko – Kharkiv National University of Radio Electronics, Postgraduate Student at the Computer Modelling and Intelligent Technologies Department; Kharkiv, Ukraine;
e-mail: oleksii.kovalenko3@nure.ua
ORCID ID: https://orcid.org/0009-0008-4779-6161

ВИКОРИСТАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ УПРАВЛІННЯ СОРТУВАННЯМ ПОСИЛОК У КОНВЕЄРНОМУ ПОТОЦІ

Предметом дослідження *є моделі прийняття рішень автоматизованих сортувальних ліній (ASL), якими оснащені перевалкові центри сортування посилок (PSC).* ***Мета роботи*** *– розроблення моделі прийняття рішень для ASL завантажувальних дверей терміналів PSC у вигляді штучної нейронної мережі (ANN), що виконує завдання сортування посилок з огляду на їх ваги й габарити. У статті розв'язано такі* ***завдання:*** *проаналізовано конструктивні особливості модифікації обладнання ASL для реалізації трьох варіантів логіки завантаження вантажівок; визначено діапазони ваги й габаритів посилок, пов'язаних з номерами завантажувальних лотків і дверей; установлено критерії для реалізації заданої логіки сортування посилок; визначено параметри наборів даних для навчання й тестування моделей штучних нейронних мереж (ANNs); з використанням інструментарію середовища MATLAB проведено навчання й тестування чотирьох видів ANNs: RBFNN, GRNN, PNN, FFNN; із застосуванням як метрики якості відносної точності правильної класифікації посилок за діапазонами їх ваги й габаритів і кількості використовуваних нейронів проведено порівняльний аналіз ANNs.* ***Методи:*** *системний, аналітичний, комп'ютерного моделювання, методи навчання з архітектурою ANN, математичний і статистичний аналіз ефективності навчання.* ***Досягнуті результати.*** *Проведено порівняльне навчання й тестування чотирьох видів ANNs: RBFNN, GRNN, PNN, FFNN. За результатами тестування для реалізації обрано модель FFNN, навчену методом Левенберга – Марквардта, який забезпечує відносну точність класифікації ОПВ на рівні 97%.* ***Висновки.*** *Розроблена модель прийняття рішень ASL у вигляді FFNN дає змогу здійснити класифікацію посилок у потоці конвеєра за трьома діапазонами їх вагогабаритних параметрів для реалізації заданої логіки сортування за завантажувальними дверима або лотками. Реалізована логіка сортування забезпечує компактне завантаження кузовів вантажівок і знижує ризики пошкодження посилок, коли вони укладаються одна на одну.*

Ключові слова: *логістика доправлення; модель прийняття рішень; класифікація; автоматизована сортувальна лінія; штучні нейронні мережі.*

Бібліографічні описи / Bibliographic descriptions

Гребеннік І. В., Коваленко О. А. Використання штучних нейронних мереж для управління сортуванням посилок у конвеєрному потоці. *Автоматизовані системи управління та прилади автоматики*. 2026. № 1 (188). С. 17–32. DOI: https://doi.org/10.30837/0135-1710.2026.188.017

Grebennik, I., Kovalenko, O. (2026), "Use of artificial neural networks for managing the sorting of parcels in a conveyor flow", *Management Information System and Devices*, No. 1 (188), P. 17–32. DOI: https://doi.org/10.30837/0135-1710.2026.188.017