ISSN 2442-6571

Human Capital Decision Intelligence (HCDI) architecture microbiology laboratory based on machine learning and operations research models

Suryasatriya Trihandaru [a,1,*], Yosia Adi Susetyo [a,2], Hanna Arini Parhusip [a,3], Bambang Susanto [a,4]

[a] Master of Data Science, Faculty of Science and Mathematics, Satya Wacana Christian University, Salatiga, Indonesia
[1] suryasatriya@uksw.edu; [2] adi.yosia1995@gmail.com; [3] hanna.parhusip@uksw.edu; [4] bambang.susanto@uksw.edu
* corresponding author

ARTICLE INFO

Article history
Received July 31, 2024
Revised September 19, 2025
Accepted September 28, 2025
Available online November 30, 2025

Keywords
Machine learning
Operations research
Personnel assignment
Human resource management
Optimization

ABSTRACT

The Human Capital Decision Intelligence (HDCI) system integrates human-computer interaction in a microbiology laboratory that uses machine learning and operational research to classify new tasks and then recommend assignments to each person. The models evaluated in building this system are Support Vector Machine, Gaussian Naive Bayes, Multinomial Logistic Regression, and Artificial Neural Network. The results of the research show that the ANN model is the most consistent and reliable across various training ratios, as indicated by the model's goodness parameters. The selected ANN model is combined with a linear programming approach to optimize workload distribution. The integrated system successfully manages new job scenarios and recommends staff based on competencies and availability. It also ensures assignments do not exceed maximum workload limits and finds alternatives when key staff are unavailable. The implementation of the HDCI system has a positive impact on various factors, including the fair distribution of tasks, enhanced staff performance monitoring, and significantly improved operational efficiency and human resource management in the microbiology laboratory. The system is designed to be easy to use and support collaboration between laboratory staff and computational models. The system is not only advanced in supporting personnel management decision-making, but it can also demonstrate how artificial intelligence and operations research systems can be combined to address the needs of the microbiology laboratory environment.

1. Introduction

Human resource management in microbiology laboratories significantly impacts operational efficiency and the quality of analysis results. Currently, the microbiology laboratory's recruitment system at one of the drug companies in Indonesia remains manual. The process starts with the administrative unit that receives the analysis request, prepares the assignment letter, and then submits the sample and assignment to the supervisor. The supervisor distributes the work to the staff and records it in the workbook, with tasks divided according to habit. This process is not optimal, causing problems such as uneven division of work, unequal workloads among staff, reduced productivity due to inefficient assignments, and staff dissatisfaction because new tasks are assigned without regard for unfinished ones. As a result, certain personnel's workload accumulates. To solve this problem, an innovative approach was undertaken using a mathematical model that combines machine learning methods with linear programming and operations research. The machine learning method is used to classify the weight of the assignment letter given to the staff. The most commonly used machine learning methods for classification are Support Vector Machine, Gaussian Naive Bayes, and Multinomial Logistic Regression [1]. In addition to conventional methods, Artificial Neural Network (ANN) technology allows developers to build network architectures to obtain accurate and optimal classification results [2]. Several studies that combine ML and OR methods, namely the James Cock 2023 research on machine learning, are used to classify environmental conditions and focus on benchmarking [3]. Furthermore, Ayoub's 2024 research on Scheduling Problems demonstrates how to combine Machine Learning (ML) and Operations Research (OR) resources to solve complex scheduling issues [4]. HDCI's research gap compared to previous research is a clear focus on human resource management in the laboratory; the system not only classifies workloads with ML but also directly optimizes the distribution of tasks using OR.

In this study, we will compare the durability of the four machine learning techniques to complete the work weight classification tasks on the task sheet, and, at the same time, we will also optimize the percentage ratio of training data and test data to best perform the classification. Using ANN techniques, a neural network architecture, called the Human Capital Decision Intelligence (HDCI) architecture, has been built, tailored to the data being handled. Each machine learning technique is measured by the goodness of a model that includes accuracy, mean squared error (MSE), root mean square error, precision, F1 score, recall, and confusion matrix [5]. The initial step in troubleshooting is compiling sample receipt data. A digital sample reception form has been built that is connected to the MySQL database so that the data can be collected and processed correctly [6]. At the processing stage, data stored for a certain period is further processed. This process aims to correct typing errors, reduce incomplete data, and overcome data redundancy, thus obtaining clean and ready-to-treat data [7]. *The Ratcliff/Obershelp* algorithm works to correct typing errors on parameter variables [8].

Data preprocessing results are used to build machine learning models and research linear programming operations. In a machine learning model, algorithms classify the work weight on a task sheet in the category of work weight points 1 (very easy), 2 (easy), 3 (medium), and 4 (complex). The weighted point result of this classification is summed up over a certain period and used as an obstacle variable in the operations research method. This variable is an indicator of the active workload of each staff member, which must not exceed the maximum workload limit specified by the superior. When the staff's workload reaches its maximum, the status changes from 'ready' to busy'; the status of 'busy' will prevent further assignments. In addition to the work weight, the operations research model also considers other impediment variables such as staff competence, job handling speed, and staff presence. The system makes optimal decisions based on existing constraints. Therefore, this study aims to develop and test a mathematical model that integrates machine learning (ML) and operations research (OR) methods to solve job assignment problems in microbiology laboratories. The entire calculation and development of the model are done using the Python programming language and the necessary library.

2. Method

Fig. 1 describes the methods used to integrate machine learning and operations research to address staffing issues in microbiology laboratories. The figure illustrates a complete machine learning pipeline that begins with fetching the dataset from a MySQL database and continues with selecting relevant features to prepare the data for analysis. The preprocessing stage includes case folding, text cleaning, and removing unnecessary spaces, and is further enhanced using the Ratcliff/Obershelp algorithm to improve input accuracy. After preprocessing, the data is structured and readied for built-in machine learning processes. Multiple classification algorithms, SVM, GNB, MLR, and ANN, are then optimized and evaluated using different training–testing ratios. Finally, the model with the highest evaluation performance is selected as the classification engine for determining task-letter weight and subsequently integrated into the operations research system.

Vol. 11, No. 4, November 2025, pp. 602-621

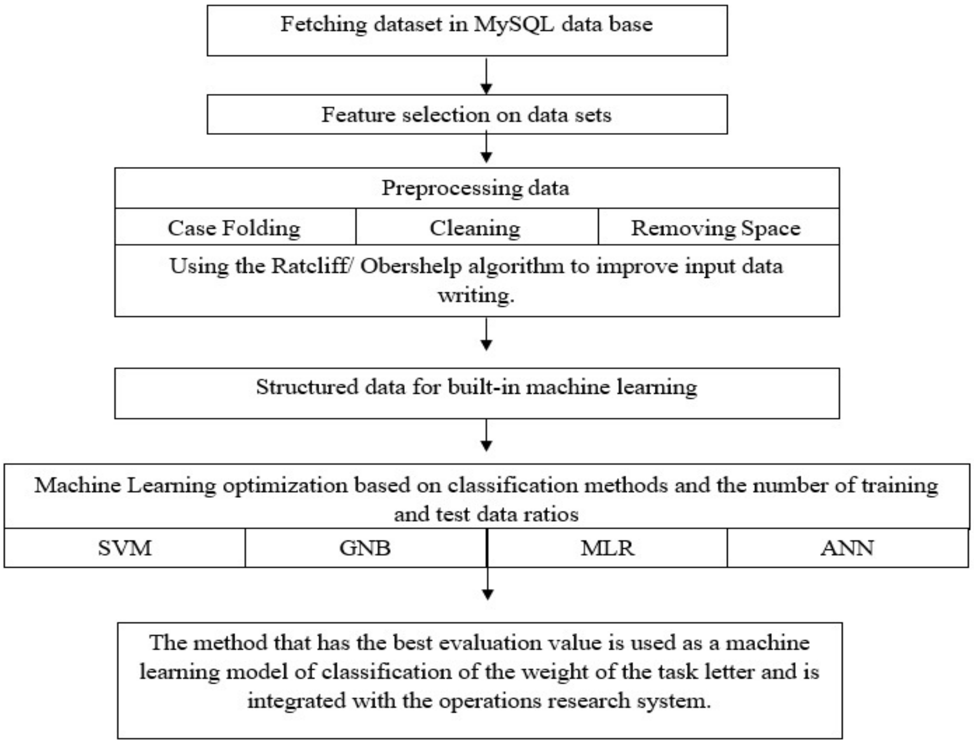


Fig. 1. Workflow integrates ML and OR

2.1. Data Recording

Data preprocessing is performed by fetching data from the MySQL database. Data was taken from the input period from September 3, 2023, to February 1, 2024. The total amount of data owned is about 13,000 input data points, with the data feeds shown in Fig. 2.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13377 entries, 0 to 13376
Data columns (total 10 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   analyst_name       13377 non-null  object
 1   parameter          13377 non-null  object
 2   sample_name        13377 non-null  object
 3   process_sample     13377 non-null  object
 4   sample_amount      13377 non-null  float64
 5   parameter_amount   13377 non-null  float64
 6   total_parameters   13377 non-null  int64
 7   days_count         13377 non-null  int64
 8   sample_arrival     13377 non-null  object
 9   sample_completion  11623 non-null  object
```

Fig. 2. The example of features's database used in this research

2.2. Preprocessing Data

Preprocessing can improve data quality, making it cleaner, more accurate, and consistent, and in the required format and dimensions, so it can be used as material for machine learning and operations research [9],[10]. In performing workload classification, the database features are the collection parameters listed in Table 1. The problem with the data is that there are typing errors and variations when referring to the same analytical parameters. The Ratcliff/Obershelp algorithm determines the similarity between two strings by matching the strings [11]. This algorithm corrects input data for a parameter variable, using the corpus containing the parameter analysis from a microbiological laboratory as the reference.

Table 1. A corps of parameter analysis capable of performing in the microbiology laboratory of the company

Analysis Parameter Type		
enterobacter	azospirillum	analisamikroskopis
pseudomonas	pelarutphospat	shigella
alt	bakteriselulolitik	totalcoliform
khamir	rhizobium	ecoli
azotobacter	salmonella	activitywater
candidaalbicans	staphylococcusaureus	trichoderma
lactobacillus	saccharomyces	bifidobacterium
bakteriasamlaktat	kapang	clostridium
mpncoliform	aspergillus	penambatnitrogen
		biletolerantgramnegatifbacteria

The Ratcliff/Obershelp algorithm works by matching the input to the corpus, with a similarity percentage determined [12]. The rate of similarity is calculated with the equation as follows:

$$\sigma = \frac{2K_m}{|S_1|+|S_2|} \times 100\% \qquad (1)$$

K_m is the same number of characters in both strings, $|S_1|$ is the length of the first string, and $|S_2|$ is the length of the *second* string. An example of the work of the Ratcliff/Obershelp algorithm in the handling of data when an error occurs in the input of the analytical parameter "SalmOneelA"; the name of the closest and most corresponding analytical parameter to the corpus is "salmonella" (Table 2). The algorithm will count the number of characters in each input that is specified as $|S_1$ | and $|S_2$ | is the total number of valid characters in the corpus.

Table 2. Search for a subsequence anchor sample

Search	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
Sring1	s	a	l	m	o	n	E	e	l	a
String2	s	a	l	m	o	n	E	l	l	a

Search for a subsequence where the longest substring is called the anchor, then look for the other substrings of both strings. Anchor1 = salmone = 7, Anchor2 =la =2. Then, combined from both substrings, we found Km= 7+2 = 9, the value $|S_1|$=10; $|S_2|$=10; $|K_m|$= 7+2=9. So, in the above case, the value of σ,i.e., $\sigma = \frac{2\,x\,9}{|10|+|10|}\,x100\,\%$ = 90 %. The calculations show that 90% of the input words correspond to the corpus of "salmonella". Therefore, on the data set, the input will be corrected to "salmonella." In this study, Ratcliff/Obershelp worked effectively in measuring string similarity in the type of analysis parameters, but this algorithm has limitations where small changes and misspellings in one parameter will have a significant impact on the similarity score so that the same paremeter can be identified differently [13].

2.3. Data structuring for machine learning

As mentioned in Chapter 1 of the machine learning model, algorithms classify the work weight on the task sheet by the categories and work weight points 1 (very easy), 2 (easy), 3 (medium), and 4 (complex). The company has 13,000 assignment letters; hence, it is necessary to use machine learning for a quick process. The process is as follows:

2.3.1. The specified variables and the number of data

Initially, we define the independent variable ($\vec{x} \in \mathbb{R}^n$) and dependent variable (y) for classification [14]. We set the data to contain A=13,000 assignment letters, n =24 := the number of parameter types; N =12;= the number of personnel. Therefore, we denote: x_i := parameter type i-th; $\vec{x}$:= the vector of a parameter type y:= the vector weight of the assignment letter denoting the classification result. This will be obtained from an optimal machine-learning process.

- Frequency definition

  The calculation of the frequency (f) of the analysis parameters that appear in the data was carried out by adopting the literature [15]. In this case, the number of word parameters in the document is adjusted, that is, $f=a/b$ with a: = the number of appearances of a parameter name in a document; b: = the number of documents. We chose that b=1 for prototyping HCDI in this research.

- Operating weight definition

  The operating weight is defined by paying attention to the frequency of work. Routine tasks are more accessible because they are performed frequently, while rare tasks require more precision and careful execution. Therefore, the weight of the work is defined as inversely proportional to its frequency. The operating weight is then normalized. We define f_i Is the i^{th} frequency, F_{max} for $\max\{f_i\}_{i=1,..,n}$, F_{min} for $\min\{f_i\}_{i=1,..,n}$, O_i is i^{th} operating weight as follows:

$$O_i = \frac{F_{max}-f_i}{F_{max}-F_{min}} \quad (2)$$

If $F_{max} = f_i$, then we set $O_i = 0.01$ since the parameter weight should not be zero, based on formula (2), we get the list of operating weights and their frequency due to the parameter names accordingly by mining the 13,000 assignment letters during the observation period, as shown in Table 3.

Table 3. Work weight on each type of work

Parameter Name (x_i)	Frequency (f_i)	Operating Weight (O_i)	Parameter Name (x_i)	Frequency (f_i)	Operating Weight (O_i)
alt	10483	0.0100000	pelarutphospat	182	0.9831075
kapang	8088	0.2285742	azotobacter	129	0.9881657
khamir	7759	0.2599733	azospirillum	129	0.9881657
salmonella	6036	0.4244131	lactobacillus	128	0.9882611
ecoli	6001	0.4277534	rhizobium	84	0.9924604
enterobacter	5986	0.4291850	bakteriselulolitik	71	0.9937011
clostridium	3680	0.6492651	aspergillus	55	0.9952281
shigella	3672	0.6500286	biletolerantgramnegatifbacteria	44	0.9962779
staphylococcusaureus	1428	0.8641916	saccharomyces	41	0.9965642
pseudomonas	1105	0.8950181	trichoderma	26	0.9979958
mpncoliform	525	0.9503722	bakteriasamlaktat	7	0.9998091
totalcoliform	211	0.9803398	candidaalbicans	5	1.0000000

Further, the 13,000 assignment letters will be classified into four categories of classes, i.e., and 1,2,3,4, representing very easy, easy, medium, and complex, respectively. We seek operating weight values on a new assignment letter based on customer requests in a vector size of n x 1 and symbolized as $\vec{O}_j$, j=1,...,A. Parameters that do not appear are filled with zero. The weight v_j of assignment letters is then obtained from the number of weight values of the parameter contained in it, i.e.

$$v_j = \sum_{i=1}^{n} o_{i,j}\ , o_{i,j} := component\ i-th\ from\ \vec{O}_j, j = 1,\dots, A\ , A = 13{,}000 \quad (3)$$

2.3.2. Labelling definition

The following process is labeling the assignment letter's weight into four classes. The labeling process is defined by the provisions shown in Table 4. Next, the machine learning classification process can be performed, with the target variables being values 1, 2, 3, and 4. Below are the SVM methods, Multinomial Logistic regression, and Gaussian Naive Bayes. The four models are used to get the best fit for the assignment letter's weight classification. Next, we need to define feature vectors and target variables. Features vectors are vectors that carry the operating weight of each assignment letter. The target variable is a variable that consists of classes 1 and 2, 3, 4, which in succession state classes very easy, easy, medium, and complex.

Table 4. Labeling of assignment letters based on the weight of work contained therein

Class	Weight value v_j	Score
Very Easy	$v_j \leq Q_1$	1
Easy	$Q_1 < v_j \leq Q_2$	2
Medium	$Q_2 < v_j \leq Q_3$	3
Complex	$v_j > Q_3$	4

Note: $Q_{1,2,3}$ are the quarter weights of the task sheet on the entire data set

2.4. Optimization Machine Learning

2.4.1. Literature Review

2.4.1.1. SVM Method

SVM (Support Vector Machine) is a classification technique. SVM algorithms in machine learning can be used for regression processes and classifying linear and nonlinear data. The SVM algorithm works by searching for separating fields, often called the best hyperplane, to separate data sets into different classes. The main characteristic of SVM is its ability to minimize empirical classification errors while maximizing geometric margins [16],[17]. Previous research used SVM techniques to perform several case classifications and obtained the accuracy values shown in Table 5.

Table 5. SVM technical accuracy in dealing with classification cases

Author	Application of SVM techniques	Accuracy
Waheed et al. [18]	Predicting Student Academic Performance from Big Data VLE	79.95% - 89.14%
Ma et al. [19]	Radiomics based on an unimproved MR sinus sequence of the heart in the early diagnosis of hypertensive heart disease	83.3%
Wang et al. [20]	Machine learning to identify the risk of high-frequency hearing impairment in the general population	82.94%
Hui & Chiew [21]	Enhanced Network Intrusion Detection Based on CNN-LSTMSA	79.95-82.94%

2.4.1.2. Gaussian Naive Bayes Method

Naive Bayes is a probability-based classification method used in machine learning. The process is based on Bayes' theorem and the assumption of feature differences in the data. In Naive Bayes, strong feature independence implies that features in the data have equal weights and are unrelated [22]. Some studies used the Naive Bayes method, and the accuracy values obtained are presented in Table 6.

Table 6. Gaussian Naive Bayes's technical accuracy in dealing with classification cases

Author	Usage of the Naive Bayes technique	Accuracy
Afdhaluzzikri et al. [23]	Classification using the Naïve Bayes algorithm for water quality datasets	93.15%-95.73%
Ferdowsi et al. [24]	Classification of vasovagal syncope from physiological signals on tilt table testing	86.1%-86.9 %
Shariati et al. [25]	Determine the visual prognosis of patients with open globe injuries using machine learning approaches	85%

2.4.1.3. Multinomial Logistic Regression Method

Logistic regression is a binary logistic model based on one or more features to estimate the probability of binary responses, such as in-layer or out-layer, pass or fail, yes or no [26]. Multinomial logistic regression is an extension of logistic regression, used when a dependent variable has more than two categories. This method helps predict the outcome of a dependent variable with several classes (more than two). The following is a previous study that used logistic regression, and its accuracy values are shown in Table 7 .

Table 7. Accuracy of logistic regression techniques in dealing with classification cases

Author	Application of logistic regression classification method	Accuracy
Lee et al. [27]	Predict User Satisfaction with Metaverse Services Through Machine Learning	74 %- 88 %
Musleh et al. [26]	Application of logistic regression to machine learning models for patients with prostate-specific antigens in grey zones	79.2 %
Zhou et al. [28]	For the prediction of pulmonary complications after thoracoscopic surgery	83.1%

2.4.1.4. Artificial Neural Network (ANN) method

An Artificial Neural Network (ANN) is a machine learning algorithm that mimics how the human brain processes information. The ANN consists of neurons arranged into three main layers: the input, hidden, and output layers. During training, the weights and biases on inter-neuronal connections are adjusted via stochastic gradient descent to reduce predictive errors. ANNs can be arranged in various architectural configurations with variations in the number of layers and types of activation functions. Once the training is complete, the ANN is tested with new data to evaluate its performance. Thus, the primary objective of ANN modeling is to determine the correct weights for each connection so that the network performs the desired function. This is achieved by multiplying each input by the corresponding weight, summing up all the inputs, and adding bias values [29]–[31]. The ANN method has proven effective in solving various classification cases, as shown in Table 8.

Table 8. ANN's technical accuracy in dealing with classification cases

Author	Use of the neural network technique	Accuracy
Reynolds et al. [32]	ANN to predict building demand for the next 24 hours and solar PV plants.	95.61 %- 98.05 %
Rau et al. [33]	Measuring, explaining, and applying complex land maps predicted uncertainty with ANN	94.37- 95.11%
Tabarzadi & Ghaemi, [34]	Predicting the load and pressure of the CO2 outlet of the independent variables of fluid, gas, and Monoethanolamine concentration rate (MEA)	99.90%

2.5. Mathematical Model in HCDI Architecture Using Operations Research and machine learning

Operation research is required here to create a mathematical model that gives the optimal result variable by considering the objective function and its constraints [35], [36]. The decision variable is a binary indicator of whether a staff member will be assigned. The objective function is to minimize the working time of the selected staff. The constraints are the maximum workload, personnel availability, and each individual's limited competence. Such an explanation forms the basis of the mathematical modeling as follows:

2.5.1. Notation in model

In this model, J_i represents the i^{th} parameter, where $i = 1,2,3, \dots, n$and $n = 24$in this case. The term $P_j$denotes the j^{th} personnel, with $j = 1,2,3,4, \dots, N$and $N = 12$. Meanwhile, t_{ij} refers to the duration required for the i^{th} job to be completed by the j^{th} personnel, as shown in (4).

$$t_{ij} = \begin{cases} > 0, & j^{th} \text{ personnel competent with } i^{th} \text{ parameter} \\ -1, & j^{th} \text{ personnel incompetent with } i^{th} \text{ parameter} \end{cases} \tag{4}$$

- Set Selected Parameters

  Not all parameters are selected, so in this case, they are limited to the set of parameters chosen, symbolized by i^s. Selected parameters occur as requested by the customer.

- Selected personnel

  Similarly, not all personnel were selected. The selection process was carried out with an association matrix, which is not discussed here. It is assumed that the personnel competency data set is known.

$$x_{ij} = \begin{cases} 1, & \text{if } J_i \neq 0,\ P_j \neq 0 \\ & 0, \text{else} \end{cases} \tag{5}$$

In this formulation, j^s denotes the index set of selected workers, while x_{ij} represents the binary decision variable for the assignment between job i and personnel j. The term w_j indicates the total workload assigned to personnel P_j, and $w_{\max}$ specifies the maximum workload limit determined by the supervisor. Meanwhile, b_{ij}refers to the work weight assigned by job $J_i$to personnel P_j, where these weights are generated from the neural network (NN) model described in a later section.

- Staff presence: a_j is the binary variable indicates the presence of personnel P_j ,i.e

$$a_j = \begin{cases} 1, \text{if } P_j \text{ present} \\ 0, \text{if } P_j \text{ not present} \end{cases} \tag{6}$$

2.5.2. Objective function and its constraint

The objective function must minimize the time used for each J_i , ∀i ∈ i^s ⊂ ℕ. This is expressed as:

$$t_{ij}^* = min\{t_{ij}: i^* \in i^s; j^* \in j^s \} \tag{7}$$

Equation (7) needs to be updated where $P_j \neq 0$ with P_j is busy, or w_{max} is fulfilled, or $a_j = 0$. Therefore, it is set x_{ij}= 0. As a result, the objective function is updated as follows: **Minimize.**

$$f_{i_{1\leq i\leq N}} = \sum_j t_{ij}\ x_{ij}\ ,\ \text{N=12} \tag{8}$$

The minimum value is denoted by f_i^* for i=i^*

2.5.3. Function of constraints

- Job selection: The J_i is assigned to P_j by minimizing $\{t_{ij}\}$,i.e.,

$$t_{ij}^* = min\{t_{ij}: j \in j^s \} \tag{9}$$

- Total workload for personnel P_j must satisfy $w_j \leq w_{max}$,i.e.,

$$\sum_{i=1}^{M} b_{ij}\, x_{ij} + w_j \leq w_{max}\ , \forall j \tag{10}$$

- The P_j can only work J_i, if P_j is competent, one yields

$$x_{ij} == 0\ \ if\ t_{i,j} = -1 \tag{11}$$

- Staff presence: P_j can only accept the assignment if P_j presents at work. In other words, (6) is satisfied

The decision variable is defined as a binary variable, as shown in (5). By solving the objective function and satisfying the constraints in (9)-(11), the optimizers are obtained to assign the work and accept the new assignment letters. The neural network architecture then improves the HCDI architecture.

2.5.4. Neural Network in the HCDI architecture

Optimization resulted in the neural network method achieving the best evaluation among other machine learning methods. This model was trained for up to 25 epochs with a batch size of 64, the

Adam optimizer, and the categorical_crossentropy loss function. Table 9 provides full details of the HDCI architecture for working weight classification.

Table 9. HDCI Neural Network Architecture for assignment letter weight classification

Layer (type)	Output Shapes	Param #	activation
dense_14 (Dense)	(None, 32)	800	Relu
dense_15 (Dense)	(None, 18)	594	Relu
dense_16 (Dense)	(None, 12)	228	Relu
dense_17 (Dense)	(None, 4)	52	Softmax
Total params: 1674 (6.54 KB); Trainable params: 1674 (6.54 KB); Non-trainable params: 0 (0.00 B).			

Fig. 3 visually shows how the Neural Network HDCI architecture classifies the weight of assignment letters.

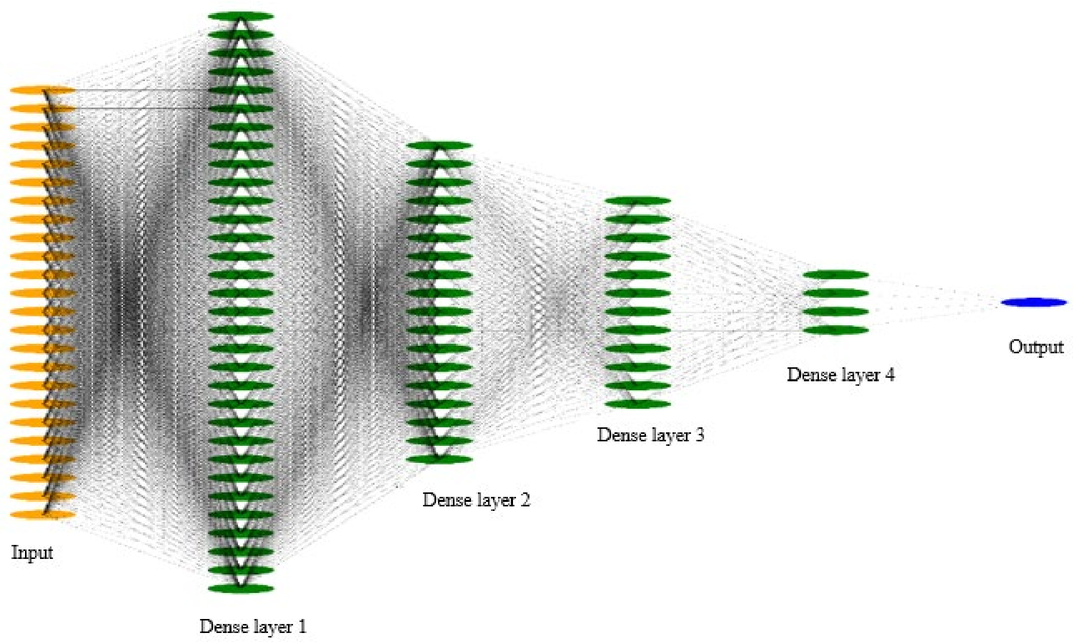


Fig. 3. HDCI Architecture with Artificial Neural Network Method in Assignment Letter Weight Classification

The result of working weight classification with a neural network is mathematically expressed as $b_{(i,j)}$ $\hat{y}$ with (y)̂ is the result of predictions from the obtained neural network. The working system of a neural network is mathematically described as follows:

- Input Layer

  The input layer, shown in Fig. 3 with orange neurons, contains the type of work described in the assignment letters. Each assignment letter is represented as an input vector, $\vec{O}_j \in \mathbb{R}^n$ with n=24 , j =1,2,3,...,A . The $\vec{O}_j$ is the weight of the work obtained from (2). This vector is the input (x) in the following NN notation. In this study, A=13,000.

- Hidden Layer

  The value obtained is z is the probability value of the end process of the NN model (Fig. 4). The maximum z-index value is chosen to indicate the assignment letter class, i.e., 1, 2, 3, or 4. The value of this class is $b_{i,j}$ in the operations research model. In other words, bi,j is the Likert scale value, where very easy=1, easy=2, medium=3, and complex=4.

Vol. 11, No. 4, November 2025, pp. 602-621

Dense layer 1 with 32 Neurons : Weight Matrix : $W_1 \in \mathbb{R}^{32 \times 24}$ Bias Vector : $b_1 \in \mathbb{R}^{24}$ Activation Function: ReLU Operation on dense layer 1 $h_1 = \text{ReLU}\ (W_1 x + b_1)$	Dense layer 2 with 18 Neurons : Weight Matrix: $W_2 \in \mathbb{R}^{18 \times 32}$ Bias Vector : $b_2 \in \mathbb{R}^{18}$ Activation Function: ReLU Operation on dense layer 2 $h_2 = \text{ReLU}(W_2 h_1 + b_2)$
Dense layer 3 with 12 Neurons : Weight Matrix: $W_3 \in \mathbb{R}^{12 \times 18}$ Bias Vector: $b_3 \in \mathbb{R}^{12}$ Activation Function : ReLU Operation on dense layer 3 $h_3 = R\text{ReLU}\ (W_3 h_2 + b_3)$	Dense layer 4 Weight Matrix: $W_4 \in \mathbb{R}^{4 \times 12}$ Bias Vector: $b_4 \in \mathbb{R}^{4}$ Activation Function: Softmax Operation on dense layer 4 $z = \text{Softmax}\ (W_4 h_3 + b_4)$

Fig. 4. Neural Network Layer Specifications

2.6. Operation research

Operation research is built on data extraction results, thus obtaining an association matrix that provides information about the time (day) spent by staff in completing each type of job. The complete data are listed in Table 10.

Table 10. The association matrix that connects the type of job to the staff in the unit of working time (days)

Parameter Analysis	Identitas Personel											
	H_An0	H_An1	H_An2	H_An3	H_An4	H_An5	H_An6	H_An7	H_An8	H_An9	H_An10	H_An11
	Average handling time (Days)											
pseudomonas	9.20	7.80	8.41	9.79	8.11	-1.00	11.23	7.42	12.00	7.16	8.40	11.34
candidaalbicans	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	12.50	-1.00	-1.00	9.00
staphylococcusaureus	11.21	7.48	8.38	10.02	-1.00	7.00	11.09	7.58	11.76	9.27	11.01	10.47
pelarutphospat	11.60	9.60	6.67	11.00	8.14	-1.00	-1.00	-1.00	13.38	7.00	12.00	11.77
bakteriasamlaktat	-1.00	7.50	16.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	12.33
lactobacillus	12.00	-1.00	7.00	7.00	8.17	-1.00	-1.00	-1.00	-1.00	-1.00	12.00	-1.00
mpncoliform	7.00	7.00	9.40	5.29	-1.00	-1.00	10.83	13.80	-1.00	-1.00	14.00	11.00
alt	9.72	7.74	8.47	10.75	18.00	7.08	10.74	6.17	10.99	9.70	10.37	10.39
clostridium	10.25	8.55	8.52	11.22	-1.00	5.97	9.26	6.36	11.05	9.76	10.43	11.24
shigella	10.25	8.54	8.54	11.22	-1.00	5.96	9.28	6.21	11.05	9.76	10.43	11.24
salmonella	10.40	8.23	8.48	11.08	8.00	6.20	10.58	6.97	11.22	9.76	10.66	10.84
kapang	10.39	8.08	8.51	11.04	18.00	6.18	10.92	7.25	11.12	9.75	10.40	10.93
khamir	10.31	8.17	8.51	11.01	18.00	6.17	10.89	7.20	11.14	9.68	10.37	10.92
azospirillum	12.00	-1.00	7.00	7.00	8.11	-1.00	-1.00	-1.00	-1.00	-1.00	12.00	-1.00
ecoli	10.28	8.30	8.48	10.99	8.00	6.22	10.35	7.07	11.09	9.85	10.69	10.80
saccharomyces	-1.00	-1.00	-1.00	7.00	7.87	-1.00	-1.00	-1.00	-1.00	-1.00	12.00	-1.00
bakteriselulolitik	-1.00	-1.00	7.00	7.00	8.37	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00
azotobacter	12.00	-1.00	7.00	7.00	8.15	-1.00	-1.00	-1.00	-1.00	-1.00	12.00	-1.00
biletolerantgramnegatifbacteria	11.00	5.67	-1.00	9.60	-1.00	-1.00	12.83	-1.00	14.33	9.25	8.67	13.82
trichoderma	-1.00	-1.00	-1.00	7.00	8.29	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00
enterobacter	9.36	8.10	8.43	11.23	-1.00	7.11	9.87	5.91	11.11	9.68	10.34	11.13
totalcoliform	-1.00	7.56	-1.00	-1.00	-1.00	-1.00	9.73	11.00	-1.00	4.00	-1.00	-1.00
rhizobium	12.00	-1.00	-1.00	7.00	8.00	-1.00	-1.00	-1.00	-1.00	-1.00	12.00	-1.00
aspergillus	12.00	-1.00	-1.00	7.00	7.90	-1.00	-1.00	-1.00	-1.00	-1.00	12.00	-1.00

While collecting personnel competency data, we acknowledge the potential for bias in the data obtained. The distribution of sample types and analysis parameters is not always balanced, of course, as some personnel often handle work at different levels of difficulty. Then, assignment data that is influenced by administrative considerations, such as work experience and availability, can also cause representation. In addition, potential errors in recording and data entry affect the reliability of the results.

2.7. HCDI Evaluation

The evaluation of the machine learning model for classifying the assignment letter's work weight is performed by comparing several techniques, including Support Vector Machine (SVM), Gaussian Naive Bayes (GNB), Multinomial Logistic Regression (MLR), and Artificial Neural Network (ANN) methods with the HDCI architecture. Four training-to-test data ratios used in this evaluation are 80:20, 70:30, 50:50, and 30:70. Each ratio is evaluated using performance metrics such as Accuracy, Mean Squared Error (MSE), Root Mean Square Error, precision, recall, and F1 score. The best machine-learning model is implemented to determine the further work weight integrated with the operating research system. With the 25 new assignment letters, the HCDI architecture delivers optimal solutions. The evaluation is done by comparing assignments with the traditional methods used in the laboratory. The best machine learning model is implemented to determine the work weight, which is subsequently integrated with the operations research system.

3. Results and Discussion

The following presents the HCDI architecture, which is experimenting with new task scenarios to see how it performs in solving distribution-of-work problems in the company's microbiology laboratory. Fig. 5 shows, in general, how the HDCI system operates in making personnel recommendations. First, the HDCI system will classify the job's weight; then it will capture the classification results and use operational research with predetermined limits to identify the optimal personnel candidates.

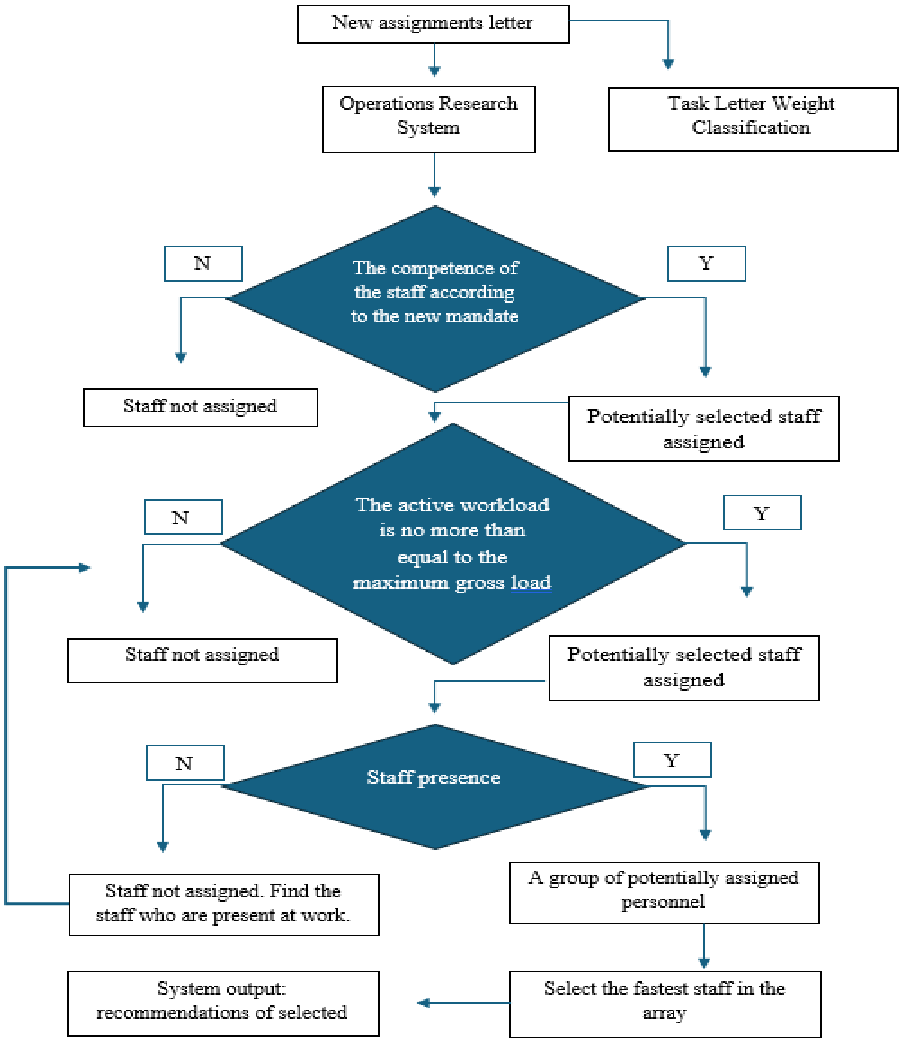


Fig. 5. Flowchart of the Hybrid Classification–Decision Integration (HCDI) Pipeline

This section presents the HCDI architecture, which is testing new task scenarios to evaluate its effectiveness in addressing work distribution issues in the company's microbiology laboratory. Fig. 4 shows, in general, how the HDCI system operates in making personnel recommendations. First, the HDCI system will classify the job weight, then capture the classification results and use operational research with predetermined limits to identify the optimal personnel candidates.

3.1. Machine Learning Optimization Results

Table 11 presents the optimization results for several machine learning models across various classification techniques, with different training and test data ratios. Comparative models include SVM, MLR, GNB, and ANN, with training-to-test ratios of 80:20, 70:30, 50:50, and 30:70. Evaluation metrics include Accuracy, MSE, RMSE, Precision, Recall, F1 Score, and Confusion Matrix. ANN consistently outperforms other models across all metrics and data splits, showing near-perfect performance with accuracy and F1 scores close to 1.00. SVM also demonstrates strong performance, particularly at higher training data ratios. MLR shows promising results but is slightly lower than SVM and ANN, especially regarding recall and F1 scores. GNB performs worst, particularly at lower training data ratios, reflected in lower accuracy and F1 scores.

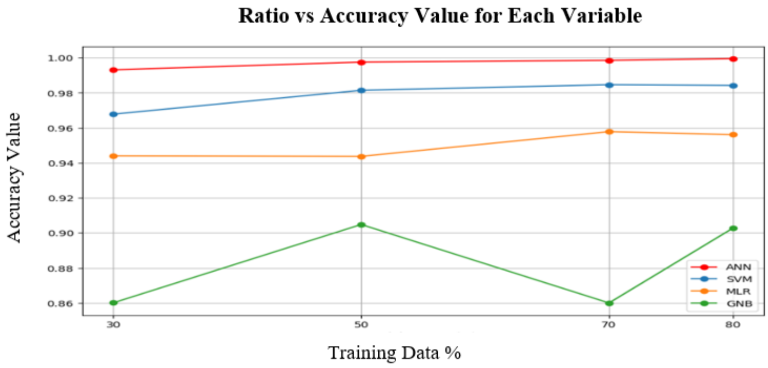
Table 11. Machine learning optimization results from various classification techniques and optimal training data to form models in which classes 1, 2, 3, and 4 are marked in succession: Class 0, Class 1, Class 2, and Class 3.

Model Name	Training data ratio: test data (%)	Accuracy (%)	MSE	RMSE	Precision (%)	Recall	F1 Score	Confusion matrix [TP,TN,FP, FN]
				SVM				
Class 0	80:20	0.9842	0.0172	0.1311	0.99	1.00	0.99	[790, 1409, 11, 0]
Class 1					0.98	0.93	0.95	[267, 1917, 6, 20]
Class 2					0.98	1.00	0.99	[1014, 1173, 18, 5]
Class 3					1.00	0.91	0.95	[104, 2096, 0, 10]
				MLR				
Class 0	80:20	0.9561	0.04525	0.2127	0.97	1.00	0.98	[789, 1396, 24, 1]
Class 1					0.90	0.79	0.84	[227, 1898,25, 60]
Class 2					0.95	0.98	0.97	[996, 1143,48, 23]
Class 3					1.00	0.89	0.94	[101, 2096, 0 , 13]
				GNB				
Class 0	80:20	0.9027	0.1122	0.33499	0.95	1.00	0.98	[789, 1381, 39, 1]
Class 1					0.71	0.86	0.78	[248, 1820, 103, 39]
Class 2					0.93	0.91	0.92	[928, 1118, 73, 91]
Class 3					1.00	0.26	0.42	[30, 2096, 0, 84]
				ANN				
Class 0	80:20	0.9995	0.0004524	0.02127	1.00	1.00	1.00	[790, 1420, 0 ,0]
Class 1					1.00	1.00	1.00	[287, 1923, 0, 0]
Class 2					1.00	1.00	1.00	[1019,1190, 1,0]
Class 3					1.00	0.99	1.00	[113, 2096, 0,1]
				SVM				
Class 0	70:30	0.9846	0.016289	0.1276	0.98	1.00	0.99	[1169, 2127, 19, 0]
Class 1					0.98	0.94	0.96	[436, 2840, 9, 30]
Class 2					0.99	0.99	0.99	[1511, 1773, 23, 8]
Class 3					1.00	0.92	0.96	[148, 3154, 0, 13]
				MLR				
Class 0	70:30	0.9578	0.0440	0.2099	0.97	1.00	0.98	[1168, 2105, 41, 1]
Class 1					0.93	0.80	0.86	[371, 2820, 29, 95]
Class 2					0.96	0.98	0.97	[1491, 1726, 70, 28]
Class 3					1.00	0.90	0.95	[145, 3154, 0, 16]
				GNB				
Class 0	70:30	0.8600	0.15897	0.39872	0.94	1.0	0.97	[1168, 2075, 71, 1]
Class 1					0.57	0.85	0.68	[395, 2552, 297, 71]
Class 2					0.93	0.82	0.87	[1243, 1701, 95, 276]
Class 3					0.98	0.28	0.43	[45, 3153, 1, 116]

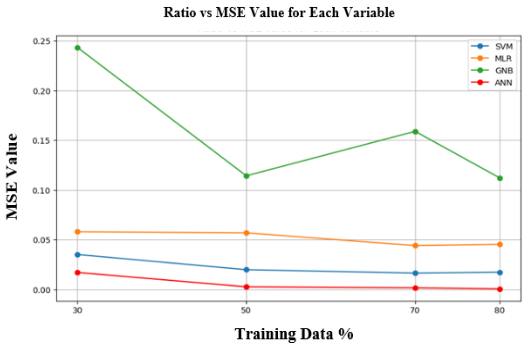
Model Name	Training data ratio: test data (%)	Accuracy (%)	MSE	RMSE	Precision (%)	Recall	F1 Score	Confusion matrix [TP,TN,FP, FN]
ANN								
Class 0	70:30	0.9985	0.001508	0.0388	1.00	1.00	1.00	[1169, 2145 , 1, 0]
Class 1					0.99	1.00	1.00	[465, 2846, 3, 1]
Class 2					1.00	1.00	1.00	[1516, 1795, 1,3]
Class 3					1.00	0.99	1.00	[160, 3154, 0, 1]
SVM								
Class 0	50:50	0.9814	0.01972	0.1405	0.98	1.00	0.99	[1980, 3508, 35,2]
Class 1					0.98	0.92	0.95	[723, 4723, 18, 61]
Class 2					0.98	0.99	0.99	[2469, 2992, 49, 15]
Class 3					1.00	0.91	0.95	[250, 5249, 1, 25]
MLR								
Class 0	50:50	0.9437	0.0568	0.2384	0.94	1.00	0.97	[1980, 3414, 129, 2]
Class 1					0.93	0.71	0.80	[555, 4697, 44, 229]
Class 2					0.95	0.98	0.96	[2440, 2904, 137, 44]
Class 3					1.00	0.87	0.93	[239, 5249, 1, 36]
GNB								
Class 0	50:50	0.9048	0.1142	0.3379	0.94	1.00	0.97	[1980, 3417, 126, 2]
Class 1					0.74	0.84	0.78	[656, 4509, 232,128]
Class 2					0.93	0.92	0.93	[2286, 2876, 165,198]
Class 3					0.96	0.28	0.48	[77, 5247, 3, 198]
ANN								
Class 0	50:50	0.9975	0.0025	0.05	1.00	1.00	1.00	[1980, 3542, 1, 2]
Class 1					0.99	1.00	1.00	[783, 4736, 5, 1]
Class 2					1.00	1.00	1.00	[2476, 3038, 3, 8]
Class 3					0.98	0.99	0.99	[272, 5245, 5, 3]
SVM								
Class 0	30:70	0.9678	0.03503	0.1872	0.98	1.00	0.99	[2785, 4897, 51, 2]
Class 1					0.97	0.83	0.89	[889, 6631, 28, 187]
Class 2					0.95	0.99	0.97	[3453, 4088, 169, 25]
Class 3					1.00	0.91	0.995	[359, 7340, 1, 35]
MLR								
Class 0	30:70	0.9440	0.0579	0.2406	0.94	1.00	0.97	[2785, 4775, 173, 2]
Class 1					0.92	0.71	0.80	[762, 4596, 63, 314]
Class 2					0.95	0.98	0.96	[3415, 4065, 192, 63]
Class 3					0.95	0.86	0.92	[340,7336, 5, 54]
GNB								
Class 0	30:70	0.8601	0.2434	0.4934	0.94	1.00	0.97	[2785, 4776, 172,2]
Class 1					0.50	0.84	0.63	[904, 5761, 898, 172]
Class 2					1.00	0.82	0.90	[2848,4246, 11, 630]
Class 3					0.99	0.29	0.45	[116, 7340, 1, 278]
ANN								
Class 0	30:70	0.9931	0.01694	0.13014	1.00	1.00	1.00	[2785, 4945, 3, 2]
Class 1					0.99	0.97	0.98	[1047, 6650, 9, 29]
Class 2					1.00	1.00	1.00	[3466, 4247, 10, 12]
Class 3					0.93	0.97	0.95	[384, 7310, 31, 10]

The results are visualized in Fig. 6, which shows that the ANN model achieves the highest and most stable accuracy across all test data ratios, followed by SVM and MLR, which also show good performance but with slight fluctuations. As shown in Table 11, Fig. 5 indicates that the GNB has the lowest accuracy, but it improves with increasing training data. GNB has the lowest evaluation due to the main limitation that the model relies on the assumption of feature independence, i.e., each predictor variable is assumed to be non-interdependent. In this study, the laboratory data used show correlations between features, such as the relationship between parameter type and the number of samples or the workload of analysts. If the independence assumption is not met, the GNB classification performance may decrease because the model tends to simplify the relationships between variables [37], [38]. One limitation of the GNB method is the balance of the data. When the number of samples in a class is much larger, the accuracy value produced appears high. Still, in a class where the minority of models show poor, less-than-optimal performance, this is especially so when the resulting model tends to be biased [39].

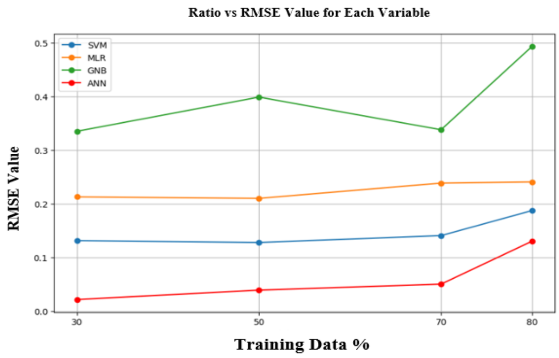
Vol. 11, No. 4, November 2025, pp. 602-621



(a)



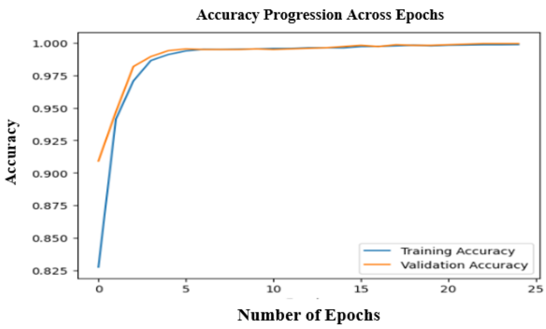
(b)



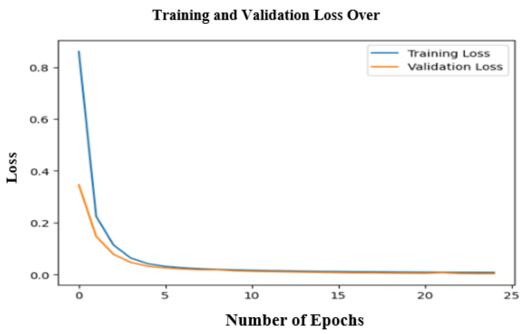
(c)

Fig. 6. Shows the evaluation chart of the accuracy model (a), MSE (b), and RMSE (c) on each epoch

The HDCI's architecture is trained for 25 epochs with a batch size of 64, the Adam optimizer, and the categorical_crossentropy loss function. The results of the training process are shown in Fig. 7a and Fig. 7b. These graphs show the accuracy of the training and validation sets for machine learning models over 25 epochs. At the start of the training, the training accuracy and the validation increased sharply, with the training precision rising from about 0.825 to almost 1,000 in the first five epochs. The same thing happened with validating accuracy, reaching nearly 1,000 over the same period. It shows that models learn quickly and achieve high performance without signs of overfitting or underfitting. This model can recognize patterns well on training data and generalize well on validation data.



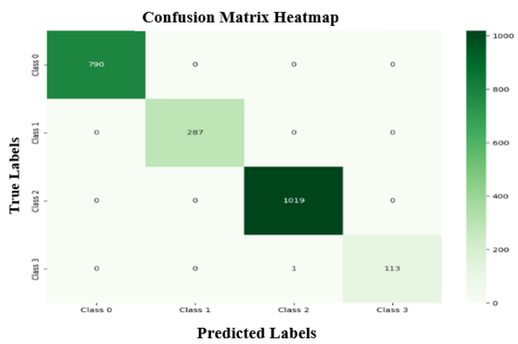
(a) Graphic accuracy during the training process and validation of machine learning models against the number of epochs



(b) Loss chart during the training process and validation of machine learning models against the number of epochs

Fig. 7. The results of the training process

Fig. 7 shows the model training results for testing the test data, along with evaluation metrics (F1-score, accuracy, MSE, RMSE, confusion matrix, precision, recall).



(a) Confusion matrix model Artificial Neural Network with HDCI architecture

```
Accuracy: 0.9995475113122172
Mean Squared Error: 0.00045248868778280545
Root Mean Squared Error: 0.021271781490575854
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       790
           1       1.00      1.00      1.00       287
           2       1.00      1.00      1.00      1019
           3       1.00      0.99      1.00       114

    accuracy                           1.00      2210
   macro avg       1.00      1.00      1.00      2210
weighted avg       1.00      1.00      1.00      2210
```

(b) Parameters of the goodness value of the machine learning model

Fig. 8. Model Training

Fig. 8a and Fig. 8b show high accuracy with only two minor predictive errors: one sample of Class 2 is predicted as Class 3, and one sample of Class 3 is forecast as Class 2. The Heatmap in Fig. 8a provides a visualisation of the prediction distribution. The diagonal element cell (from top left to bottom right) represents the number of accurate predictions for each class outside of it, indicating prediction errors. The darker green colour indicates a higher number of correct predictions. Fig. 8b shows excellent model performance, with precision, recall, and F1 scores almost perfect across all classes, in line with the confusion matrix, which shows only 2 prediction errors. It affirms the high accuracy of models in predicting data classes. The ANN model with the HCDI architecture is applied to the operations research system because it provides the best evaluation results. In this experiment, it shows that the ANN architecture has a better approach when compared to conventional machine learning methods, but it is undeniable that ANN is the initial stage in the implementation of deep learning, so when compared to the Transformer architecture in terms of performance and resolution of sequential dependencies and long contexts, ANN is still under Transformer. However, we not only see the selection of ANN in the experiment from one hand but also from the perspective of computing capabilities that are lighter, faster, lower cost in the training process, and more efficient compared to the Transformer architecture [40].

3.2. Results of NN implementation with HCDI architecture and 25 new assignment letter scenarios

The NN model with the HCDI architecture is integrated into the linear programming operations research system at equations (4)-(11) using 25 new assignment letter scenarios. In equation (10), set w_{max} =10. The operational research system aims to obtain staff recommendations based on competence, presence, speed of execution, and current workload. The implementation results are shown in Table 11. The operational research system works well in providing staff recommendations within the limits set by the method, i.e., the competence that corresponds to the assignment, the fastest completion time, the availability of staff, and the maximum workload each staff member can carry. Staff with a working weight w_j over ten w_j points ≥ w_{max} will not be given a new assignment. Staff who reach the workload limit on the table are marked with a thick red printed identity and are not included in subsequent assignment optimization. The system can also search for alternative staff when selected staff are declared absent. The final simulation has conducted experiments in which H_An9 is not present. The H_An4 staff is chosen as an alternative when H_AN9 is absent (shown in the last row of Table 12). It can't be done manually as fast as before this research.

We conclude that the HDCI architecture integrated with operational research proved capable of finding alternatives to other personnel that meet the competence requirements, do not exceed the workload threshold, and have the fastest delivery time. Conventional methods use manual recording, where staff assignments do not consider the current workload, and it is challenging to determine the weight of each assignment letter. This research showed that models with an HCDI architecture and operations research were efficient and fast, producing recommendations in just 2 minutes. On the application, new assignment letters can reach 150 per day, with a maximum limit of 50. Therefore, this research can be continued for more significant new assignments to test the model's reliability.

Table 12. A new task letter scenario is inserted into an integrated machine learning operations research system with data from September 3, 2023 to February 1, 2024, covering about 13,000 data points

Parameters in the assignment	Classification Results	Work weight of ML classification results	Max. working weight (given)	Work weight running	Selected staff ID
ALT, Kapang, Khamir, MPN Coliform	easy	2	10	2	H_An5
ALT, Kapang, Khamir, MPN Coliform, E Coli	medium	3	10	5	H_An5
ALt, Kapang, Khamir, Salmonella, Staphylococcus a, Pseudomonas a. E Coli	complex	4	10	9	H_An5
ALT, Kapang, Khamir, E Coli, Enterobacter, Salmonella, Shigella, Clostridia	medium	3	10	12	H_An5
ALT, Kapang, Khamir, E Coli, Enterobacter, Salmonella, Shigella, Clostridia	medium	3	10	3	H_An7
alt, kapang	very easy	1	10	4	H_An7
ALT, Enterobacter	very easy	1	10	5	H_An7
ALT, E.coli, Kapang, Khamir, Salmonella, Staphylococcus	medium	3	10	8	H_An7
Alt, Kapang, Khamir, E Coli, Salmonella, Pseudo , Staphylococus Aureus	complex	4	10	12	H_An7
ALT, E.coli, Enterobacter, Kapang, Khamir, Salmonella, Shigella, Clostridium	medium	3	10	3	H_An1
ALT, Kapang, Enterobacter	easy	2	10	5	H_An1
ALT, Kapang, Khamir, MPN Coliform, E Coli, Salmonella.	medium	3	10	8	H_An1
alt	very easy	1	10	9	H_An1
Kapang, khamir	easy	2	10	11	H_An1
Alt, Kapang, Khamir	easy	2	10	2	H_An2
ALT, Kapang, Enterobacter	easy	2	10	4	H_An2
Alt, Kapang, Khamir, E Coli, Salmonella, P seudo , Staphylococus Aureus	complex	4	10	8	H_An2
ALT, E.coli, Enterobacter, Kapang, Khamir, Salmonella, Shigella, Clostridium	medium	3	10	11	H_An2
Alt, Staphylo Aureus, Pseudomonas	easy	2	10	2	H_An9
Alt, Kapang	very easy	1	10	3	H_An9
Alt, Kapang, Khamir, E Coli, Salmonella, Enterobacter, Shigella, Clostridia	medium	3	10	6	H_An9
Alt, Kapang, Khamir	easy	2	10	8	H_An9
Alt, Kapang	very easy	1	10	9	H_An9
ALT, Enterobacter	very easy	1	10	1	H_An0
ALt, Kapang, Khamir, Salmonella, Staphylococcus a, Pseudomonas a. E Coli.	complex	4	10	13	H_An9
Simulation when H_An9 ID is not present					H_An4

We compared Decision Support Systems (DSS) in Hongly's (2025) previous research and Clinical Decision Support Systems (CDSS) in N. Peiffer-Smadja et al.'s (2025) research against the new HDCI system. We showed that vocational DSS emphasizes big data analysis and multicriteria modeling to evaluate student literacy, but does not lead to operational actions [41]. Furthermore, clinical CDSS emphasizes continuous adoption and usability, ensuring effective use by general practitioners, but lacks a machine learning component for prediction or optimization [42]. The HCDI system differs from the two in that it provides a complete pipeline: competency prediction (ANN), linear-programming-based work distribution, and human-computer interaction to maintain transparency and flexibility.

4. Conclusion

This article presents the development of human-computer interaction in the microbiology laboratory through prototyping the Human Capital Decision Intelligence (HCDI) architecture. The technique is the integration of machine learning with operations research. The built-in system can accurately classify the new assignment letter. The selected machine learning model uses artificial neural network techniques. The evaluation graph shows that the ANN method is more stable and improves as the amount of training data increases in each division of the training data ratio. The evaluation results showed the goodness of the model, with a parameter accuracy of 0.99954, MSE 0.0004525, RMSE 0.02127, precision 1.0 in four classes, recall 1.0 in class (0.1.2), recall 0.99 in the third class, and parameter F1 with a score of 1.0 in the four classes. The results of this evaluation show the robustness of the neural network model for conducting the classification. The HCDI architecture leads to processing the result of the assignment classification. The HCDI has been tested with 25 new job letter scenarios. The system can perform assignments well and recommend staff based on competence and the fastest time to handle the job. The following scenario shows that the system can assign staff who do not have an active workload exceeding the limits set by the superior and recommend other staff who are competent and ready to receive new assignments. The HCDI architecture has also been shown to find alternative staff when the selected staff are not on the job, as seen in the final scenario of the operations research test. This indicates that human-computer interaction provides efficiency in the microbiology laboratory, as we expected. These systems can improve job satisfaction by equitably distributing burdens, optimize resource allocation in laboratories, and potentially be expanded to more complex environments, such as hospitals or the pharmaceutical industry.

Acknowledgment

We would like to express our gratitude to the Ministry of Education, Culture, Research, and Technology for providing full funding and support for the implementation of this research.

Declarations

Author contribution. In this research, ST is the head of research, ensuring the direction and objectives go as planned. HAP analyzes results, provides insight, and interprets data deeply. YAS is building systems, collecting data, and conducting research activities directly at the research site.

Funding statement. The source of funding for this research comes from a grant from the Ministry of Education, Culture, Research, and Technology with numbers 001/LL6/PB/AL.04/2024,027/SPK-PFR/RIK/6/2024

Conflict of interest. In the publication of this work, the author stated that there is no potential conflict of interest. Furthermore, all ethical aspects, including plagiarism, informed consent, violations, data fabrication and falsification, and publication procedures, have been addressed in accordance with applicable ethical standards.

Additional information. No additional information is available for this paper.

Data and Software Availability Statements

The following GitHub page: https://github.com/Yosia2023/Data_jurnal contains dataset snippets, application software image attachments, test results, software performances, and test results for handling new assignment letters in various scenarios.

References

[1] T. Islam *et al.*, "Predictive modeling for breast cancer classification in the context of Bangladeshi patients by use of machine learning approach with explainable AI," *Sci. Rep.*, vol. 14, no. 1, pp. 1–17, 2024, doi: 10.1038/s41598-024-57740-5.

[2] N. H. Haron, R. Mahmood, N. M. Amin, A. Ahmad, and S. R. Jantan, "An Artificial Intelligence Approach to Monitor and Predict Student Academic Performance," *J. Adv. Res. Appl. Sci. Eng. Technol.*, vol. 44, no. 1, pp. 105–119, 2025, doi: 10.37934/araset.44.1.105119.

[3] J. Cock, D. Jiménez, H. Dorado, and T. Oberthür, "Operations research and machine learning to manage risk and optimize production practices in agriculture: good and bad experience," *Curr. Opin. Environ. Sustain.*, vol. 62, p. 101278, 2023, doi: 10.1016/j.cosust.2023.101278.

[4] A. Ouhadi, Z. Yahouni, and M. Di Mascolo, "Integrating machine learning and operations research methods for scheduling problems: A bibliometric analysis and literature review," *IFAC-PapersOnLine*, vol. 58, no. 19, pp. 946–951, 2024, doi: 10.1016/j.ifacol.2024.09.155.

[5] G. M. Rao, D. Ramesh, V. Sharma, A. Sinha, M. M. Hassan, and A. H. Gandomi, "AttGRU-HMSI: enhancing heart disease diagnosis using hybrid deep learning approach," *Sci. Rep.*, vol. 14, no. 1, pp. 1–19, 2024, doi: 10.1038/s41598-024-56931-4.

[6] S. Patel, S. Kumar, S. Katiyar, R. Shanmugam, and R. Chaudhary, "MongoDB Versus MySQL: A Comparative Study of Two Python Login Systems Based on Data Fetching Time," *Res. Intell. Comput. Eng.*, vol. 1254, pp. 57–64, 2020, doi: 10.1007/978-981-15-7527-3_6.

[7] A. F. Hassan, S. Barakat, and A. Rezk, "Towards a deep learning-based outlier detection approach in the context of streaming data," *J. Big Data*, vol. 9, no. 1, 2022, doi: 10.1186/s40537-022-00670-8.

[8] E. Kalbaliyev and S. Rustamov, "Learning Algorithms with Character-Based Similarity," *Digit. Interact. Mach. Intell.*, vol. 9–10, pp. 11–19, 2020, doi: 10.1007/978-3-030-74728-2.

[9] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Glob. Transitions Proc.*, vol. 3, no. 1, pp. 91–99, Jun. 2022, doi: 10.1016/j.gltp.2022.04.020.

[10] M. Z. Al-Taie, S. Kadry, and J. P. Lucas, "Online data preprocessing: A case study approach," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 4, pp. 2620–2626, 2019, doi: 10.11591/ijece.v9i4.pp2620-2626.

[11] B. Y. Ong, R. Wen, and A. N. Zhang, "Data blending in manufacturing and supply chains," *Proc. - 2016 IEEE Int. Conf. Big Data, Big Data 2016*, pp. 3773–3778, 2016, doi: 10.1109/BigData.2016.7841047.

[12] W. Hidayat, E. Utami, and A. D. Hartanto, "Effect of Stemming Nazief Adriani on the Ratcliff/Obershelp algorithm in identifying level of similarity between slang and formal words," *2020 3rd Int. Conf. Inf. Commun. Technol. ICOIACT 2020*, pp. 22–27, 2020, doi: 10.1109/ICOIACT50329.2020.9331973.

[13] M. Groen-Xu, G. Bös, P. A. Teixeira, T. Voigt, and B. Knapp, "Short-term incentives of research evaluations: Evidence from the UK Research Excellence Framework," *Res. Policy*, vol. 52, no. 6, p. 104729, 2023, doi: 10.1016/j.respol.2023.104729.

[14] S. Cohen, *The basics of machine learning: strategies and techniques*. Elsevier Inc., 2021, doi: 10.1016/B978-0-323-67538-3.00002-6.

[15] J. Jiang and K. Srinivasan, "MoreThanSentiments: A text analysis package[Formula presented]," *Softw. Impacts*, vol. 15, no. December 2022, p. 100456, 2023, doi: 10.1016/j.simpa.2022.100456.

[16] N. D. Lynn and A. W. R. Emanuel, "Using Data Mining Techniques to Predict Students' Performance. a Review," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1096, no. 1, p. 012083, 2021, doi: 10.1088/1757-899x/1096/1/012083.

[17] Q. A. B. K. Zaman, W. N. S. B. W. Yusoff, and Q. B. B. A. Shah, "Sentiment Analysis on The Place of Interest in Malaysia," *J. Adv. Res. Appl. Sci. Eng. Technol.*, vol. 43, no. 1, pp. 54–65, 2025, doi: 10.37934/araset.43.1.5465.

[18] H. Waheed, S. U. Hassan, N. R. Aljohani, J. Hardman, S. Alelyani, and R. Nawaz, "Predicting academic performance of students from VLE big data using deep learning models," *Comput. Human Behav.*, vol. 104, 2020, doi: 10.1016/j.chb.2019.106189.

[19] Z. P. Ma *et al.*, "A study on the application of radiomics based on cardiac MR non-enhanced cine sequence in the early diagnosis of hypertensive heart disease," *BMC Med. Imaging*, vol. 24, no. 1, pp. 1–9, 2024, doi: 10.1186/s12880-024-01301-9.

[20] Y. Wang, X. Yao, D. Wang, C. Ye, and L. Xu, "A machine learning screening model for identifying the risk of high-frequency hearing impairment in a general population," *BMC Public Health*, vol. 24, no. 1, pp. 1–14, 2024, doi: 10.1186/s12889-024-18636-1.

[21] B. Hui and K. L. Chiew, “An Improved Network Intrusion Detection Method Based On CNN-LSTM-SA,” *J. Adv. Res. Appl. Sci. Eng. Technol.*, vol. 44, no. 1, pp. 225–238, 2025, doi: 10.37934/araset.44.1.225238.

[22] A. F. Rochim, R. Kusumastuti, and I. P. Windasari, “Comparison of Feature Selection for Naive Bayes Classification Method in A Case Study of the Corona virus Lockdown,” *2021 Int. Conf. Data Sci. Its Appl. ICoDSA 2021*, pp. 215–220, 2021, doi: 10.1109/ICoDSA53588.2021.9617471.

[23] A. Afdhaluzzikri, H. Mawengkang, and O. S. Sitompul, “Perfomance analysis of Naive Bayes method with data weighting,” *SinkrOn*, vol. 7, no. 3, pp. 817–821, 2022, doi: 10.33395/sinkron.v7i3.11516.

[24] M. Ferdowsi *et al.*, “Classification of vasovagal syncope from physiological signals on tilt table testing,” *Biomed. Eng. Online*, vol. 23, no. 1, pp. 1–22, 2024, doi: 10.1186/s12938-024-01229-9.

[25] M. M. Shariati *et al.*, “Development, comparison, and internal validation of prediction models to determine the visual prognosis of patients with open globe injuries using machine learning approaches,” *BMC Med. Inform. Decis. Mak.*, vol. 24, no. 1, pp. 1–14, 2024, doi: 10.1186/s12911-024-02520-4.

[26] D. A. Musleh *et al.*, “Twitter arabic sentiment analysis to detect depression using machine learning,” *Comput. Mater. Contin.*, vol. 71, no. 2, pp. 3463–3477, 2022, doi: 10.32604/cmc.2022.022508.

[27] S. H. Lee, H. Lee, and J. H. Kim, “Enhancing the Prediction of User Satisfaction with Metaverse Service Through Machine Learning,” *Comput. Mater. Contin.*, vol. 72, no. 3, pp. 4983–4997, 2022, doi: 10.32604/cmc.2022.027943.

[28] C. M. Zhou, Q. Xue, H. J. Li, J. J. Yang, and Y. Zhu, “A predictive model for post-thoracoscopic surgery pulmonary complications based on the PBNN algorithm,” *Sci. Rep.*, vol. 14, no. 1, pp. 1–8, 2024, doi: 10.1038/s41598-024-57700-z.

[29] S. Liu, R. Chang, J. Zuo, R. J. Webber, F. Xiong, and N. Dong, “Application of artificial neural networks in construction management: Current status and future directions,” *Appl. Sci.*, vol. 11, no. 20, 2021, doi: 10.3390/app11209616.

[30] D. Z. Mohammed, “Optical Add-Drop Multiplexers: Enhancing High Transmission Bit Rates in Next-Generation Communication Networks,” *J. Adv. Res. Appl. Sci. Eng. Technol.*, vol. 43, no. 1, pp. 251–262, 2025, doi: 10.37934/araset.43.1.251262.

[31] M. Soori, B. Arezoo, and R. Dastres, “Artificial neural networks in supply chain management, a review,” *J. Econ. Technol.*, vol. 1, no. October 2023, pp. 179–196, 2023, doi: 10.1016/j.ject.2023.11.002.

[32] J. Reynolds, M. W. Ahmad, Y. Rezgui, and J. L. Hippolyte, “Operational supply and demand optimisation of a multi-vector district energy system using artificial neural networks and a genetic algorithm,” *Appl. Energy*, vol. 235, no. October 2018, pp. 699–713, 2019, doi: 10.1016/j.apenergy.2018.11.001.

[33] K. Rau, K. Eggensperger, F. Schneider, P. Hennig, and T. Scholten, “How can we quantify, explain, and apply the uncertainty of complex soil maps predicted with neural networks?,” *Sci. Total Environ.*, vol. 944, no. June, p. 173720, 2024, doi: 10.1016/j.scitotenv.2024.173720.

[34] P. Tabarzadi and A. Ghaemi, “Modeling and optimization of CO2 capture in spray columns via artificial neural networks and response surface methodology,” *Case Stud. Chem. Environ. Eng.*, vol. 10, no. June, p. 100783, 2024, doi: 10.1016/j.cscee.2024.100783.

[35] M. L. Bynum *et al.*, *Pyomo – Optimization Modeling in Python*, 3rd ed., vol. 67, pp. 1-27. Mexico: Springer, 2020, doi: 10.1007/978-3-030-68928-5_5.

[36] M. Löppenberg, S. Yuwono, M. R. Diprasetya, and A. Schwung, “Dynamic robot routing optimization: State–space decomposition for operations research-informed reinforcement learning,” *Robot. Comput. Integr. Manuf.*, vol. 90, no. February, p. 102812, 2024, doi: 10.1016/j.rcim.2024.102812.

[37] O. Peretz, M. Koren, and O. Koren, “Naive Bayes classifier – An ensemble procedure for recall and precision enrichment,” *Eng. Appl. Artif. Intell.*, vol. 136, no. PB, p. 108972, 2024, doi: 10.1016/j.engappai.2024.108972.

[38] M. Hajihosseinlou, A. Maghsoudi, and R. Ghezelbash, “A semi-supervised approach for mineral prospectivity mapping via weighted positive-unlabeled learning and tree-structured parzen estimator for hyperparameter optimization,” *Ore Geol. Rev.*, vol. 185, no. December 2024, p. 106783, 2025, doi: 10.1016/j.oregeorev.2025.106783.

[39] W. Wang, L. Yan, F. Liu, and Y. Li, "Improving Gaussian Naive Bayes classification on imbalanced data through coordinate-based minority feature mining," *PeerJ Comput. Sci.*, vol. 11, p. e3003, 2025, doi: 10.7717/peerj-cs.3003.

[40] N. Aydın, O. A. Erdem, and A. Tekerek, "Comparative Analysis of Traditional Machine Learning and Transformer-based Deep Learning Models for Text Classification," *Politek. Derg.*, vol. 28, no. 2, pp. 445–452, 2025, doi: 10.2339/politeknik.1469530.

[41] H. Li, "Machine learning optimization for vocational literacy education evaluation : A big data-powered decision support system," *Alexandria Eng. J.*, vol. 129, no. August, pp. 1258–1271, 2025, doi: 10.1016/j.aej.2025.08.029.

[42] N. Peiffer-Smadja *et al.*, "Determinants of sustainable adoption in primary care of a clinical decision support system for antimicrobial prescribing: A qualitative study," *Infect. Dis. Now*, vol. 55, no. 7, p. 105157, 2025, doi: 10.1016/j.idnow.2025.105157.