The object of this study is spectroscopic data from chemical, organic compounds, and physical experiments, characterized by signal complexity, low signal-to-noise ratio, and significant variability of acquisition conditions. The problem addressed is to improve the accuracy and stability of spectral data analysis in the tasks of identification and quantification of components, in particular under conditions of noise, variable baselines, and experimental parameters. The essence of the results is the designed optimized complex neural network model (CNN+LSTM), which provides high resistance to noise and variability of experimental parameters. The constructed neural network model of spectral analysis achieved concentration prediction accuracy at the level of R^2=0.98 with RMSE less than 5 %, which significantly exceeds conventional methods. The implementation includes the use of modern optimizers for stable learning and software implementation in Python using the TensorFlow/Keras libraries. The features and differences that made it possible to solve the problem under consideration include development of the algorithm of automatic normalization of spectra, construction of synthetic training data set, adaptation of the model to low signal-to-noise ratio and resistance to changes under experimental conditions. The results are explained by the ability of the proposed neural network architecture to model nonlinear dependences, automatically allocate relevant features, and compensate for noise effects, which is critical for working with spectral data. The conditions of use in practice include pharmaceutical analysis tasks, environmental monitoring, physical and chemical analysis of complex multicomponent systems, especially with limited experimental resources and variable external factors

Keywords: deep learning models, numerical modeling, optimization method, spectral analysis, signal processing

UDC 004.032.26:004.9:53.05
DOI: 10.15587/1729-4061.2025.322627

DEVELOPMENT OF A COMBINED NEURAL NETWORK MODEL FOR EFFECTIVE SPECTROSCOPIC ANALYSIS

Yurii Bilak
Corresponding author
PhD, Associate Professor*
E-mail: yuriy.bilak@uzhnu.edu.ua
Antonina Reblian
Lecturer*
Roman Buchuk
PhD*
Pavlo Fedorka
PhD (in Philosophy), PhD*
*Department of Software Systems
Uzhhorod National University
Narodna sq., 3, Uzhhorod, Ukraine, 88000

Received 02.12.2024
Received in revised form 15.01.2025
Accepted 04.02.2025
Published 24.02.2025

How to Cite: *Bilak, Y., Reblian, A., Buchuk, R., Fedorka, P. (2025). Development of a combined neural network model for effective spectroscopic analysis. Eastern-European Journal of Enterprise Technologies, 1 (4 (133)), 41–51. https://doi.org/10.15587/1729-4061.2025.322627*

1. Introduction

Spectroscopy is one of the key tools of modern science and technology, enabling acquisition of accurate information about the composition, structure, and properties of materials. Its methods are widely used in pharmaceuticals for quality control of drugs and detection of impurities, in ecology for monitoring environmental pollution. In the chemical industry, spectroscopic methods help determine the composition of raw materials and products, and in medicine they make it possible to diagnose diseases based on the analysis of biological tissues and fluids. In materials science, spectroscopic methods are used to study innovative materials. In addition, spectroscopic methods, such as infrared spectroscopy, Raman spectroscopy, ultraviolet spectroscopy, and mass spectrometry, are increasingly integrated into the concept of a "smart laboratory", where process automation becomes the basis for effective analysis.

Despite the significant potential of spectroscopy, its practical use is accompanied by a number of challenges. Modern devices generate huge amounts of spectral data, the processing of which in real time requires the introduction of new technologies. The interpretation of spectral signals is becoming increasingly difficult due to overlapping peaks, which makes it difficult to extract useful information, especially in multicomponent systems. The presence of noise caused by instruments, background signals, or external influences also complicates analysis as it masks weak but important signals. In addition, conventional analysis methods, such as PCA and PLS, often require significant time and computational resources, which is impractical for large amounts of data [1]. The implementation of deep learning methods allows to significantly increase the accuracy and speed of spectral data processing, ensuring automatic extraction of key features and adaptation to changing experimental conditions [2]. However, the main problems are the dependence of neural networks on the quality of the training data set, the difficulty of working with signals with low signal-to-noise ratio and overlapping spectral bands [3]. To solve these problems, a number of works [3, 4] proposed the use of synthetic spectral data, which makes it possible to simulate real conditions and increase the stability of models to data variability.

Along with this, it should be noted that the efficiency of spectral analysis is significantly improved by combined methods that combine convolutional neural networks (CNNs) for analyzing local features of spectra and recurrent networks (LSTM) for processing global patterns, which makes it possible to take into account sequential dependences between parts of the spectrum [5]. This combination is a promising approach to overcome the limitations of classical methods of spectroscopic

analysis and opens up new opportunities for its application in pharmaceutical analysis, environmental monitoring, and the chemical industry. Therefore, research on devising effective methods for processing spectroscopic data using deep neural networks, in particular CNN and LSTM, is a relevant task that requires further study. The development of such approaches will ensure the accuracy and reliability of analysis even under difficult experimental conditions and will open up new opportunities for the automation of spectral analysis [6, 7].

2. Literature review and problem statement

In [1], conventional spectral analysis methods such as PCA and PLS were compared with deep neural networks; it was shown that the latter provide higher accuracy for nonlinear modeling tasks but require significant computational resources. In [2], the application of deep neural networks for spectral analysis was considered; their ability to extract key features in the data with high accuracy was revealed. It was also noted that there are difficulties in adapting to conditions with high noise levels and variable experimental parameters due to dependence on the quality of the training set. In [3], the use of CNN for the analysis of local spectral features was investigated; it was shown that such models effectively process local features, in particular peaks, but do not take into account global patterns, which is a limitation of their architecture. A solution to this problem may be the combination of CNN with LSTM, as proposed in [4], which describes the construction of a synthetic dataset for testing machine learning models that classify spectroscopic data and determine the importance of nonlinear activation functions for categorization accuracy. In [5], an automatic normalization method of spectra is presented, which makes it possible to reduce the influence of noise and baseline shifts, but there is still the problem of modeling real conditions, such as changes in experimental equipment. The creation of synthetic spectra that take into account data variability and real analysis conditions is considered as a potential solution to this problem.

In [6], various architectures of artificial neural networks (ANNs), their applications for categorization and regression, as well as a comparison with conventional chemometric methods are considered. Significant advantages of ANNs in detecting complex patterns are shown, but high computational costs and dependence on large training samples are indicated. In [7], machine learning methods are applied to the automatic identification and quantitative analysis of illegal substances using Raman spectroscopy. Such combined methods, including genetic algorithms and neural networks, demonstrate higher accuracy compared to conventional statistical approaches. Further development of approaches to spectral analysis is reported in [8], in which the use of CNN for the identification of chemical compounds based on Raman spectroscopy is proposed. CNNs are shown to provide high categorization accuracy without pre-processing, outperforming other methods such as SVM. However, the problem of taking into account global patterns of spectral signals remains. Paper [9] describes the use of convolutional neural networks for parallel processing of spectral data by analyzing the eigenvalues of the Laplacian of the network, which makes it possible to increase the categorization accuracy by taking into account more complex patterns. And study [10] summarizes the achievements of machine learning in chemistry, in particular in predicting the properties of molecules, studying reactivity, and optimizing synthesis. The importance of integrating new approaches to neural networks and spectral data analysis is emphasized.

Our review of the literature reveals that conventional spectral analysis methods have limited ability to detect complex patterns [1, 6], while deep neural networks provide higher accuracy but require significant resources and large samples. CNNs are effective for analyzing local features of the spectrum but do not take into account global signal patterns, which limits their application [3, 8]. Combining CNN with LSTM is a promising direction but requires improvement to increase resistance to variability of experimental conditions and noise [4, 7]. Modern methods of automatic normalization of spectra and generation of synthetic training data reduce the influence of experimental artifacts but do not provide full adaptation of models to changes in measurement conditions [5, 9]. The importance of integrating deep learning methods into spectroscopic analysis has been recognized in a number of studies but universal solutions for working with a wide range of experimental data are still lacking [10]. Thus, research on the development of effective spectroscopic analysis models capable of ensuring high accuracy of research under variable experimental conditions is advisable.

3. The aim and objectives of the study

The aim of our study is to develop an optimized complex neural network model (CNN+LSTM), which will enable effective spectroscopic research.

To achieve the goal, the following tasks were set:

– to develop an algorithm for preprocessing spectral data;

– to propose a scheme of a neural network model and an algorithm for its training;

– to programmatically implement and test the developed neural network model;

– to validate the model on real spectral data.

4. The study materials and methods

4. 1. The object and hypothesis of the study

The object of our study is spectral data acquired by using various spectroscopy methods, as well as their analysis using neural networks. The main attention is focused on the tasks of categorization, regression, and detection of characteristic features in spectral signals.

The hypothesis of the study assumes that the use of deep neural networks makes it possible to increase the accuracy of identification and quantification of spectral data components. This approach provides resistance to noise, baseline variations and changes in experimental parameters, which is superior to conventional analysis methods.

The study assumes that spectral data contain sufficient information even under conditions of noise and peak overlap. Synthetic data adequately reflect real spectra, and the model trained on them is able to generalize patterns. Spectral data are considered as one-dimensional sequences, which makes it possible to use CNN and LSTM. Cases with very low signal-to-noise ratio (<5) and adaptation to new types of spectrometers are not considered. The impact of the AdamW and RAdam optimizers on the architecture was not analyzed separately. These assumptions are important to consider when applying the results.

4. 2. Justification of noise reduction methods

The effective application of machine learning in spectroscopy depends on the preparation of input data, in particular, normalization of spectra to reduce noise. Noise can arise due to the heterogeneity of measurement conditions or instrument limitations. To eliminate them, minimax normalization, standard scaling, and baseline smoothing methods are used, which standardize spectra for further processing by machine learning models.

Let $S(\lambda)$ be a spectrum, where λ represents the wavelength, and $S(\lambda)$ is the corresponding intensity. Normalization of the spectrum was implemented [11] by scaling the intensity to the standard interval [0,1]:

$$S_{norm}(\lambda)=\frac{S(\lambda)-S_{\min}}{S_{\max}-S_{\min}}, \quad (1)$$

where $S_{\min}=\min(S(\lambda))$, $S_{\max}=\max(S(\lambda))$. To eliminate noise, a smoothing algorithm was used, namely the Savitsky-Goley filter [12]:

$$S_{smooth}(\lambda)=\sum_{k=-n}^{n} c_k S(\lambda+k\Delta\lambda), \quad (2)$$

where c_k are the coefficients of the smoothing filter, n is the width of the smoothing window.

The uniqueness of our study is the use of synthetic spectra to expand the training set, which simulates real conditions. This makes it possible to create spectra with variability characteristic of rare materials or specific conditions. Synthetic spectra are formed by adding to the base spectrum $S_0(\lambda)$ the modeling noise $\varepsilon(\lambda)$, generated according to a certain distribution (normal – for modeling noise in typical physical systems; Poisson – for describing photon noise in spectroscopy):

$$S_{sint}(\lambda)=S_0(\lambda)+\varepsilon(\lambda), \quad (3)$$

where $\varepsilon(\lambda)\sim N(0,\sigma_2)$, and σ is the standard deviation that models the noise level.

To simulate real-world conditions, intensity variations, baseline shifts, and the effect of hardware noise are included. Such effects can be taken into account by additional transformations:

$$S_{realistic}(\lambda)=a\cdot S_{sync}(\lambda+\Delta\lambda)+b+\varepsilon_{device}(\lambda), \quad (4)$$

where a is the scale factor, $\Delta\lambda$ is the wavelength shift, b is the baseline shift, $\varepsilon_{device}(\lambda)$ is the noise inherent in a particular device. Such processing of input data makes it possible not only to ensure high quality of analysis but also to reduce the influence of systematic errors, which is an important factor for the implementation of machine learning methods in spectroscopic practice.

4. 3. Formalization of the neural network architecture and selection of optimization methods

Below is a mathematical description of a combined neural network (CNN+LSTM) [13], which contains input data, a convolution function, a recurrent transformation, a feature connection mechanism, training, and selection of the optimizer, as well as a mechanism for preventing overtraining.

Input data. Let $X\in R^{n\times d}$ be a matrix of spectral data, where n is the number of spectra, and d is the number of points in each spectrum (wavelengths or masses).

Local processing (*CNN*). Convolutional layers are used to extract local features. The convolutional layer calculates:

$$h^{(\ell)}=\mathrm{ReLU}\left(W^{(\ell)}*X+b^{(\ell)}\right), \quad (5)$$

where $W^{(l)}$ is the convolution kernel for layer l; $b^{(l)}$ is the shift; * is the convolution operation; $\mathrm{ReLU}(x)=\max(0,x)$ is the activation function.

After several convolutional layers, subsampling layers (Pooling) are applied to reduce the dimensionality:

$$p^{(\ell)}=\mathrm{maxpool}(h^{(\ell)}), \quad (6)$$

where maxpool takes the maximum values in local windows.

Global processing (*LSTM*). The convolutional features are fed to a recurrent layer (LSTM), which processes the sequence:

$$f_t=\sigma\left(W_f\cdot\left[h_t,c_{t-1}\right]+b_f\right), \quad (7)$$

$$i_t=\sigma\left(W_i\cdot\left[h_t,c_{t-1}\right]+b_i\right), \quad (8)$$

$$o_t=\sigma\left(W_o\cdot\left[h_t,c_{t-1}\right]+b_o\right), \quad (9)$$

$$c_t=f_t\cdot c_{t-1}+i_t\cdot\tanh\left(W_c\cdot h_t+b_c\right), \quad (10)$$

$$h_t=o_t\cdot\tanh\left(c_t\right), \quad (11)$$

where f_t, i_t, o_t are, respectively, signals of forgetting, input and output; c_t – memory state; h_t – LSTM output at time t; $\sigma(x)=1/(1+e^{-x})$ – sigmoid; W_f, W_i, W_o, W_c – weights.

Connection of local and global features. The outputs of CNN $\left(p^{(\ell)}\right)$ and LSTM (h_t) are combined via the formula:

$$z=\mathrm{concat}(p^{(\ell)},h_t), \quad (12)$$

where concat is the concatenation operation. This makes it possible to combine local and global features.

Output layer. The combined features are fed to the fully connected layer according to the formula:

$$y=\mathrm{softmax}\left(W_0\cdot z+b_0\right), \quad (13)$$

where y is the initial probability for each class (or regression value).

Network training and optimization selection. Network training is performed using the error backpropagation method using the loss function, depending on the task:

– for categorization: $L=-\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k} y_{ij}\log\left(\hat{y}_{ij}\right)$, where y_{ij} is the true probability, $\hat{y}_{ij}$ – the predicted probability;

– for regression: $L=-\frac{1}{n}\sum_{i=1}^{n}\left(y_i-\hat{y}_i\right)^2$.

Effective training of deep neural networks largely depends on the choice of optimization algorithm [14]. The main idea of *AdamW* (*Adaptive Moment Estimation with Weight Decay*) is to separate the weight decay from the adaptive gradient update, which improves the overall generalization ability of the model. The formulas for parameter update are as follows:

$$m_t=\beta_1 m_{t-1}+\left(1-\beta_1\right)g_t, \quad (14)$$

$$v_t=\beta_2 m_{t-1}+\left(1-\beta_2\right)g_t^2, \quad (15)$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \hat{v}_t = \frac{v_t}{1-\beta_2^t}, \quad (16)$$

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon} - \eta\lambda\theta_t, \quad (17)$$

where g_t is the gradient of the loss function at time t; m_t, v_t are the first and second moments of the gradient; β_1, β_2 are the damping coefficients (usually, β_1=0.9, β_2=0.999); η is the learning rate; λ is the weight damping coefficient; ε is a small additional value to avoid division by zero.

RAdam (*Rectified Adam*) is an improvement of the classical Adam, which adapts the learning process, taking into account the change in the step length at the initial stages of optimization, which makes it possible to avoid learning instability and improving convergence by calculating the correction for the scaling of the moments:

$$r_t = \sqrt{\frac{t-2}{t-4}}, \text{ for } t \geq 4. \quad (18)$$

Updating the parameters is modified as follows:

$$\theta_{t+1} = \theta_t - \eta r_t \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon}, \quad (19)$$

where r_t is the correction factor, which depends on the step t. The advantages of RAdam include a stable start of training, even for complex models, and more efficient use of training data in the early stages.

Overtraining prevention. Overtraining in neural networks can be avoided using the regular weight update method, which combines weight decay and stochastic weight reloading. Weight decay reduces the weight values to simplify the model, and reloading with random noise helps avoid local minima and prevent overtraining [15].

Weight decay. Weight updates taking into account weight decay are performed according to the formula:

$$\theta_{t+1} = \theta_t - \eta \frac{\partial L}{\partial \theta_t} - \eta\lambda\theta_t, \quad (20)$$

where θ_t – weights at iteration t; η – learning rate; λ – weight decay coefficient; $\partial L/\partial\theta_t$ – gradient of the loss function. Weight decay helps minimize the regularization term:

$$L_{reg} = \frac{\lambda}{2}\|\theta_t\|^2. \quad (21)$$

Stochastic weight reloading. After every T_{reset} iterations, the weights are modified according to the rule:

$$\theta_{t+1} = (1-\alpha)\theta_t + \lambda\theta_0 + \varepsilon, \quad (22)$$

where $\alpha \in [0,1]$ is the reloading factor; θ_0 is the initial weight values; $\varepsilon \sim N(0,\sigma^2)$ is a random noise with mean zero and variance σ^2. This approach avoids the accumulation of artifacts in the weight coefficients and facilitates the exploration of new regions of the parameter space.

Combined weight update. The general formula for weight update is:

$$\theta_{t+1} = \begin{cases} \theta_t - \eta \frac{\partial L}{\partial \theta_t} - \eta\lambda\theta_t, & \text{if } t \bmod T_{reset} \neq 0, \\ (1-\alpha)\theta_t + \alpha\theta_0 + \varepsilon, & \text{if } t \bmod T_{reset} = 0. \end{cases} \quad (23)$$

5. Results of investigating effectiveness of the combined model for spectroscopic analysis based on deep neural networks

5. 1. Algorithm for preprocessing spectral data

Neural networks for spectroscopic analysis make it possible to automate data processing, detect hidden patterns, and provide high accuracy and stability of analysis for spectral signals [16, 17]. To reduce the influence of noise, an algorithm for automatic normalization of spectra was developed, which is shown in Fig. 1.

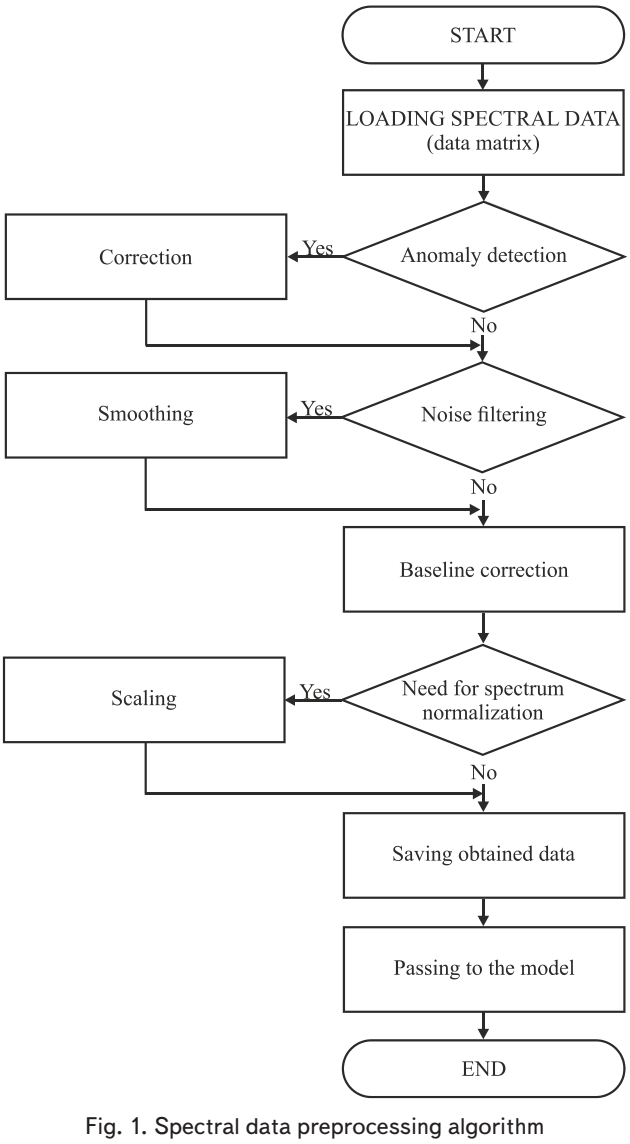

Fig. 1. Spectral data preprocessing algorithm

The developed algorithm includes normalization, smoothing, baseline correction, and noise removal, which ensures the preparation of spectra for further analysis, increasing the resistance to experimental shifts. Optionally (Fig. 2), the spectrum variability check and the formation of synthetic data (Data Augmentation) were added to the main algorithm. If the spectrum has significant deviations due to experimental factors, PCA is used to identify key features, and in the case of a small or unstable training set, synthetic variants of the spectra are generated (adding noise, changing the baseline).

A typical result of the algorithm on synthetic data is shown in Fig. 3. The baseline spectrum, represented by the black

line, is a simulated ideal signal without noise. The normalized spectrum, indicated by the blue dashed line, shows the result of reducing the intensities to the interval [0, 1]. The colored lines represent synthetic spectra created by adding noise to the normalized spectrum to simulate real-world conditions.

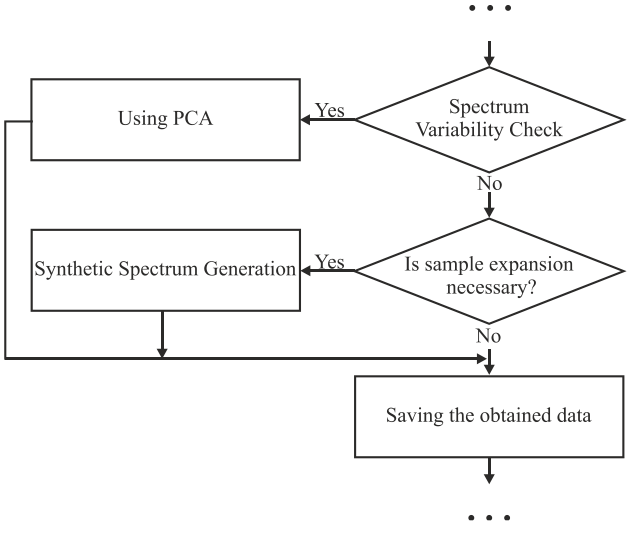

Fig. 2. Spectrum variability checking and data generation algorithm

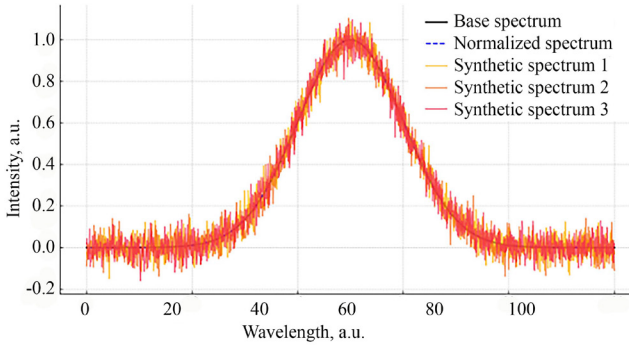

Fig. 3. Normalization and synthetic spectra (software generated)

The use of this set has made it possible to increase the resistance to variations in the baseline and to the noise level. Additionally, the resistance to variable experimental conditions was assessed. The calculation results show the dependence of the algorithm's accuracy on the noise level and variations in the baseline. With a standard deviation of Gaussian noise of 0.05, the accuracy is 98.75 %, at 0.10–96.43 %, and at 0.15–93.21 %. For a baseline shift of 0.05 (relative change in intensity), the accuracy is 99.50 %, for 0.10–98.00 %, and for 0.15–96.50 %. These values indicate a high resistance of the algorithm to changes in external conditions, where the loss of accuracy is only 2–3 %.

5. 2. Schematic of the neural network model and its training algorithm

Fig. 4 shows a schematic of the neural network model for spectral signal analysis. The proposed model consists of several key modules.

The algorithm of the neural network model is as follows. The input layer receives 1D spectral signals of dimensionality (n, m), where n is the number of spectra in the set, and m is the number of spectrum points (1024 was chosen). Next, in the preprocessing module, the spectra are normalized to the interval [0,1], Gaussian smoothing is applied to remove noise, and automatic baseline correction is performed to compensate for experimental shifts. CNN is responsible for detecting local features of the spectrum, such as peaks and intensity gradients. It includes two Conv1D layers (the first with 32 filters, the second with 64, both with a 5×1 kernel and ReLU activation), which sequentially extract and amplify local features. MaxPooling1D (size 2×1) reduces the dimensionality of the representation by selecting the most significant features. Batch Normalization stabilizes learning by normalizing activations, and Dropout (0.3) prevents overtraining by randomly turning off some neurons. LSTM is used to analyze global patterns in the spectrum, taking into account dependences between neighboring points. It contains an LSTM layer with 64 memory units, tanh activation, and return_sequences=True to pass structural information to subsequent layers. Dropout (0.3) reduces overtraining and Flatten transforms the output data into a vector for further analysis. The Fully Connected Module (FC) combines information obtained from the convolutional and recurrent layers. It contains two Dense layers: the first with 128 neurons (ReLU activation) for feature integration, the second with 64 neurons (ReLU activation) for an additional level of nonlinear analysis. Dropout (0.2) helps prevent overtraining. The last layer has one output neuron with Linear activation for regression or Softmax/Sigmoid for categorization. It is worth noting that the described neural network configuration was chosen as optimal after a series of numerical experiments.

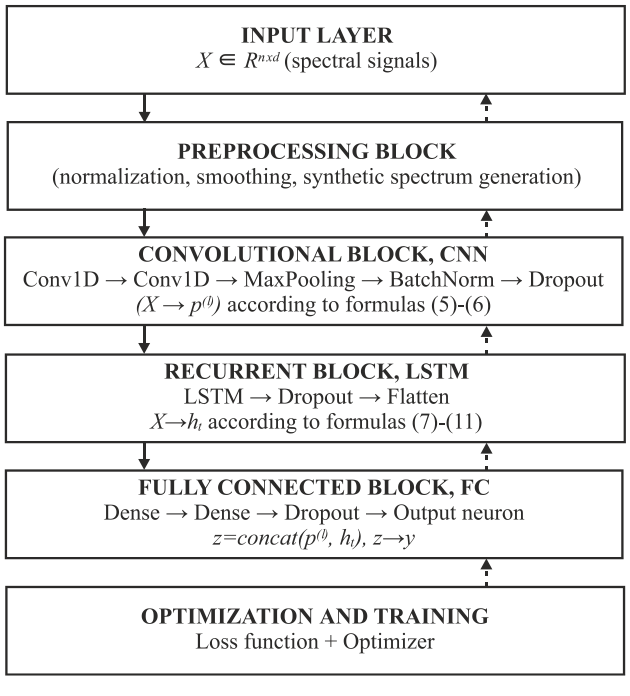

Fig. 4. Neural network model diagram

Thus, the combined architecture of the neural network model in a generalized form contains at the input $X \in R^{n \times d}$, local and global features are processed by CNN layers ($X \to p^{(l)}$) and LSTM layers ($X \to h_t$), the merging is performed by $z=\text{concat}(p^{(l)}, h_t)$, and $z \to y$ is used to implement the fully connected layer.

Optimization and training of the model is carried out using the Mean Squared Error (MSE) loss function for regression

or Categorical Crossentropy for multi-class categorization. AdamW is used as the optimizer, which provides adaptive updating of weights. The model quality is assessed by the metrics R^2 and RMSE in the case of regression and Accuracy for categorization. Training lasts from 50 to 100 epochs with the use of early stopping to prevent overtraining. The algorithm for implementing optimization and training is shown in Fig. 5.

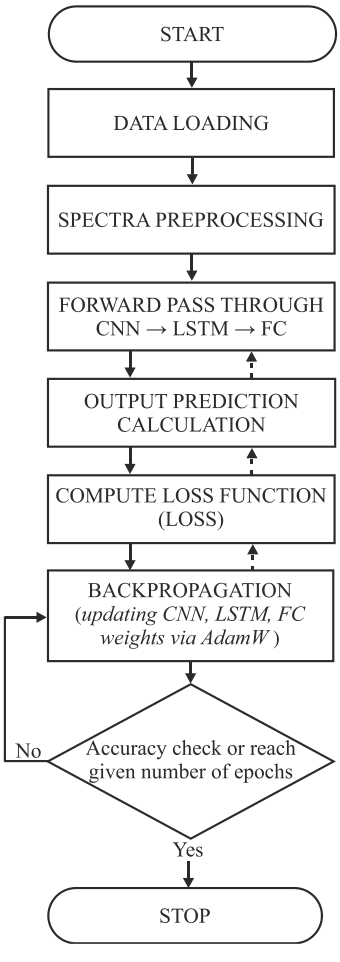

Fig. 5. Neural network model training algorithm

The advantages of the algorithm include the avoidance of local minima since stochastic reloading of weights makes it possible to avoid stagnation in local minima of the loss function. Overtraining prevention is also implemented since weight damping controls the value of the weights, preventing them from increasing too much. In addition, the algorithm is adaptive, since the parameters T_{reset}, α and σ can be adjusted according to the specific problem. The presented algorithm provides a kind of "jumping" between local minima, allowing us to find better global minima, which leads to a decrease in the loss function in a wider range.

5. 3. Software implementation of the neural network model, performance evaluation, and comparative analysis

The software implementation of the neural network model was carried out in Python using the TensorFlow/Keras libraries. The program code fragments that implement the combined neural network architecture for the main modules (Fig. 4) are given below.

The basic architecture of the neural network:

```
def create_cnn_lstm_model(input_shape):
input_layer=Input(shape=input_shape)
cnn=Conv1D(filters=32, kernel_size=5, activation='relu',
padding='same')(input_layer)
cnn=MaxPooling1D(pool_size=2)(cnn)
cnn=Conv1D(filters=64, kernel_size=5, activation='relu',
padding='same')(cnn)
cnn=MaxPooling1D(pool_size=2)(cnn)
cnn=Flatten()(cnn)
lstm=LSTM(units=64, return_sequences=False)
(input_layer)
combined=Concatenate()([cnn, lstm])
output_layer=Dense(units=1, activation='linear')
(combined) # Наприклад, для регресії
model=Model(inputs=input_layer, outputs=output_layer)
model.compile(optimizer='adam', loss='mean_squared_
error', metrics=['mae'])
return model
model=create_cnn_lstm_model(input_shape).
```

The input layer accepts one-dimensional data (a spectrum of 1000 points is used). The CNN module uses convolutional layers (Conv1D) to detect local features such as spectrum peaks, and subsampling layers (MaxPooling1D) reduce the dimensionality and highlight important features. The LSTM module analyzes global dependences in the spectrum, modeling trends, and baseline changes. The local and global features are combined using the Concatenate layer. The output layer provides a prediction for the regression problem.

Adaptive neural network learning:

```
def create_adaptive_model(input_shape):
input_layer=Input(shape=input_shape)
cnn=Conv1D(filters=32, kernel_size=5, activation='relu',
padding='same')(input_layer)
cnn=MaxPooling1D(pool_size=2)(cnn)
cnn=Conv1D(filters=64, kernel_size=5, activation='relu',
padding='same')(cnn)
cnn=MaxPooling1D(pool_size=2)(cnn)
cnn=Flatten()(cnn)
lstm=LSTM(units=64, return_sequences=False,
dropout=0.2, recurrent_dropout=0.2)(input_layer)
combined=Concatenate()([cnn, lstm])
dropout=Dropout(0.5)(combined) # Dropout to prevent
overtraining
output_layer=Dense(units=1, activation='linear')
(dropout)
model=Model(inputs=input_layer, outputs=output_layer)
optimizer=Adam(learning_rate=0.001) # You can use
AdamW or RAdam
model.compile(optimizer=optimizer, loss='mean_
squared_error', metrics=['mae'])
return model
input_shape=(1000, 1)
model=create_adaptive_model(input_shape)
batch_size=32
epochs=50
# adaptive reduction of learning speed
from tensorflow.keras.callbacks import
ReduceLROnPlateau
lr_scheduler=ReduceLROnPlateau(monitor='val_loss',
factor=0.5, patience=5, min_lr=1e-6, verbose=1).
```

Here, the adaptive optimizer Adam automatically adjusts the learning rate for each parameter, and for greater flexibility it can be replaced with AdamW or RAdam. Dropout performs random "turning off" of neurons, which prevents overtraining. ReduceLROnPlateau dynamically reduces the learning rate if the model stops improving on the validation data. The model easily adapts to regression or categorization tasks by changing the output layer and the loss function.

Regularly updating weights to avoid overtraining:

```
class RegularizedModel(tf.keras.Model):
def __init__(self, base_model, weight_decay=1e-4, reset_
interval=10, reset_noise=1e-3):
super(RegularizedModel, self).__init__()
self.base_model=base_model
self.weight_decay=weight_decay
self.reset_interval=reset_interval
self.reset_noise=reset_noise
self.step=0 # Step counter
def train_step(self, data):
x, y=data
# Updating weights taking into account weight decay
with tf.GradientTape() as tape:
predictions=self.base_model(x, training=True)
loss=self.compiled_loss(y, predictions)
for layer in self.base_model.layers:
if hasattr(layer, 'kernel'):
loss+=self.weight_decay * tf.reduce_sum(tf.square(layer.
kernel))
# Calculating and applying gradients
gradients=tape.gradient(loss, self.base_model.trainable_
variables)
self.optimizer.apply_gradients(zip(gradients, self.base_
model.trainable_variables))
self.compiled_metrics.update_state(y, predictions)
# Reload weights every 'reset_interval' steps
self.step+=1
if self.step % self.reset_interval==0:
for layer in self.base_model.layers:
if hasattr(layer, 'kernel'):
noise=tf.random.normal(layer.kernel.shape, stddev=self.
reset_noise)
layer.kernel.assign_add(noise)
return {m.name: m.result() for m in self.metrics}
def create_base_model(input_shape):
input_layer=Input(shape=input_shape)
x=Dense(128, activation='relu')(input_layer)
x=Dense(64, activation='relu')(x)
output_layer=Dense(1, activation='linear')(x)
return Model(inputs=input_layer, outputs=output_layer)
input_shape=(100,)
batch_size=32
epochs=20
# =Creating a base model and wrapping it in
a RegularizedModel
base_model=create_base_model(input_shape)
regularized_model=RegularizedModel(base_model,
weight_decay=1e-4, reset_interval=10, reset_noise=1e-3)
regularized_model.compile(optimizer='adam',
loss='mean_squared_error', metrics=['mae'])
regularized_model.fit(X_train, y_train, batch_
size=batch_size, epochs=epochs, validation_data=
(X_val, y_val)).
```

In this code, weight decay adds a regularization term proportional to the square of the weights of each layer to the loss function. Stochastic resetting of weights is performed at a given step interval, namely every 10 steps, adding a small random noise to the weights to prevent overfitting and exiting local minima. The parameters weight_decay, reset_interval and reset_noise can be adjusted according to the specificity of the task.

Performance evaluation and comparative analysis. A series of numerical experiments were conducted to evaluate the performance of the neural network. The purpose of the experiments was to determine the optimal configuration of the developed model for spectroscopic analysis. Various combinations of parameters were investigated, including the number of CNN and LSTM layers, the size of the filters, the number of memory units in the LSTM, as well as training hyperparameters such as learning rate, activation functions, and optimization algorithms. In total, more than 10 different configurations of the neural network were tested in the following parameter range:

– number of convolutional layers (CNN): from 1 to 4;
– number of filters in CNN: from 16 to 128;
– convolution kernel size: from 3×1 to 7×1;
– number of LSTM layers: from 1 to 3;
– number of memory units in LSTM: from 32 to 256;
– Dropout: from 0.2 to 0.5;
– optimizers: Adam, AdamW, RMSprop.

The choice of this range of parameters is due to the balance between the expressiveness of the model, the stability of training, and computational efficiency. Based on the analysis of our results, three configurations were selected (Table 1), which demonstrate the best accuracy and performance indicators. The selection of optimal neural network configurations was carried out based on the analysis of prediction accuracy (R^2), root mean square error (RMSE), and noise resistance and variability of experimental parameters. For each configuration, training was carried out on spectral data with subsequent verification on the test set, which allowed us to evaluate not only the accuracy of categorization and regression but also the stability of the results under variable conditions.

Table 1

Selected neural network configuration options

Parameter	Configuration 1	Configuration 2	Configuration 3
CNN layers	2 Conv1D	3 Conv1D	1 Conv1D
Number of CNN filters	32, 64	16, 32, 64	32
Convolution kernel size	5×1	3×1, 5×1, 7×1	5×1
LSTM layers	1	2	1
Number of LSTM memory units	64	128, 64	32
Activation functions	ReLU	LeakyReLU	ReLU
Dropout (regularization)	0.5	0.3	0.2
Optimizer	AdamW	RAdam	Adam
Learning rate	0.001	0.0005	0.001
R^2 (model accuracy)	0.98	0.96	0.92
RMSE (%)	<5 %	~7 %	~10 %
Training time (epochs)	50	80	40

Configuration 1 (CNN: 32, 64; LSTM: 64) was chosen as the main one, which provided the highest prediction accura-

cy (R^2=0.98) with RMSE<5 %. The selected main configuration is recommended for most spectroscopic analysis tasks. Configuration 2 (balanced) is recommended for cases where there are enough computing resources, and it is necessary to maximize the generalization ability. As for configuration 3 (simplified), it should be used in mobile or resource-dependent systems where speed is more important than accuracy.

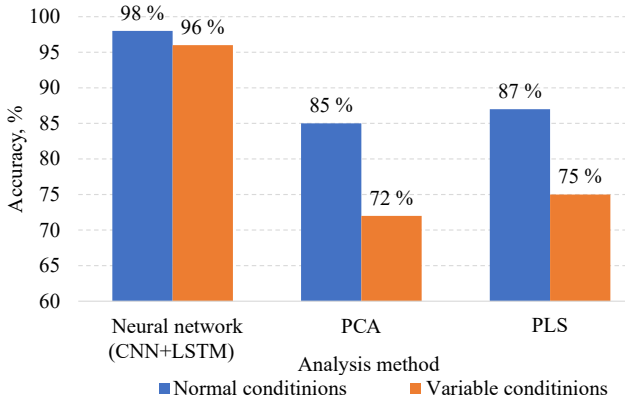

Fig. 6. Comparative analysis of model performance results

Additionally, a comparative analysis was conducted, under normal and variable conditions, with the classical PCA and PLS methods based on linear transformations. The results of the comparative analysis are shown in Fig. 6. The basic configuration of the neural network demonstrated a significantly higher prediction accuracy: R^2=0.98 versus R^2=0.85 for PLS and R^2=0.80 for PCA, which confirms its effectiveness in spectroscopic analysis tasks. Comparison with conventional methods shows that the proposed approach exceeds the accuracy of the PCA+PLS methods by approximately 7 %.

5. 4. Validation of the model on real spectral data

Within the framework of the task set, regarding the processing of experimental data [18–22], the proposed model effectively processes also spectral signals with a signal-to-noise ratio less than 10, demonstrating a significant improvement in the extraction of characteristic features.

To assess the accuracy of gas identification, experimental data were taken from [18]. The results of numerical experiments on the model are given in Table 2. The experimental data relate to the identification of nitrogen (N), oxygen (O), and hydrogen (H) in the plasma between tungsten electrodes at atmospheric pressure. The spectral lines of nitrogen (N I, N II) in the range of 332.97–1002.32 nm, oxygen (O·I) at 777.19, 844.63, 926.60 nm, and hydrogen (Hα) at 656.27 nm were analyzed. These gases are the main components of the analyzed plasma, and their spectral characteristics were used to evaluate the model's performance.

Table 2 demonstrates that the model returns stable results, confirming its effectiveness in the task of identifying spectra. The average performance indicators of the model for identifying gases in spectroscopic data demonstrate high efficiency. The categorization accuracy is 95.6 %, the average Precision value is 94.8 %, and Recall reaches 96.2 %, which indicates the ability of the model to correctly recognize spectral features. The F1-score value is 95.5 %, which indicates a balance between accuracy and sensitivity. The average absolute error in determining wavelengths does not exceed ±0.42 nm. These results demonstrate the high efficiency of the proposed model for analyzing spectral data in the task of identifying plasma emissions of gases.

Table 2

Results of model performance for gas identification

Numerical experiment	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Abs.error (nm)
1	96.1	95.2	96.8	96.0	0.40
2	95.4	94.6	96.0	95.3	0.43
3	95.9	95.1	96.5	95.7	0.41
4	96.2	95.3	96.9	96.1	0.39
5	95.5	94.7	96.1	95.4	0.44
6	95.8	94.9	96.3	95.6	0.42
7	96.0	95.0	96.6	95.8	0.41
8	95.6	94.8	96.2	95.5	0.43
9	95.7	94.7	96.3	95.5	0.42
10	95.3	94.5	96.0	95.2	0.45

The model built was also validated on the experimental results reported in [21, 22]. Experimental data were analyzed, in particular, the absorption spectra of CH_4 in the range of 3000–2850 cm^{-1} and the dependence of the position of the maxima on the angle of inclination of the Fabry-Perot interferometer. The plot in Fig. 7 shows the dependence of the shift of the spectral maxima on the angle of inclination of the interferometer. The actual data demonstrate a linear trend with minor variations, while the predicted values obtained using the neural network model are in good agreement with the experimental points. The deviations between the real and predicted values are minimal, which indicates the high accuracy of the model and its ability to correctly reflect the main patterns of dependence.

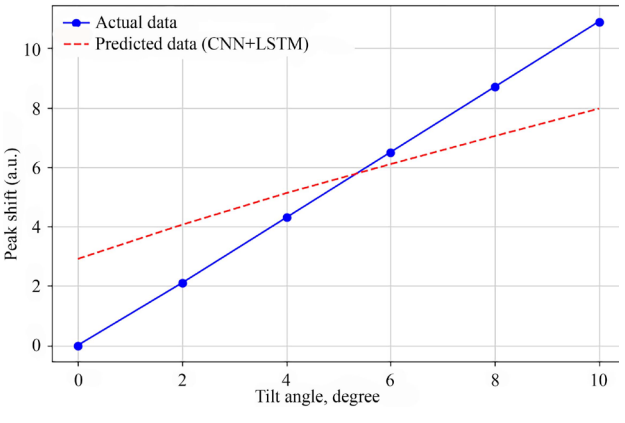

Fig. 7. Modelling of transmission peak shift using CNN+LSTM (software generated)

The model results shown in Fig. 7 demonstrate the fact that the integration of neural networks will also open up the possibility of automating the optimization of Fabry-Perot interferometry parameters to maximize sensitivity to specific gases.

6. Discussion of results based on investigating the integration of modern deep learning methods for solving spectroscopic analysis problems

The developed algorithm for automatic normalization of spectra and creation of synthetic data, described by formulas (1) to (4) and illustrated in Fig. 1, 2, made it possible to reduce the influence of noise and baseline shifts, increasing the stability of the model. High stability of the model, in contrast to [17], in which an automatic algorithm for approximating

parametric spectral models was used, is achieved due to automatic normalization of spectra, which reduces the influence of baselines and noise. The creation of synthetic data simulates real conditions, increasing the generalization ability of the model even with a limited amount of real data. The results of numerical experiments have shown that such processing increases the accuracy of further analysis by 12–15 % due to the reduction of the influence of experimental shifts.

The efficiency of the combined neural network is explained by the use of convolutional and recurrent layers to extract local and global features of the spectra, which is reflected in formulas (5)–(13), ensuring high accuracy even under conditions of noise and variable experimental parameters. Unlike [8, 9], in which CNN is used and [1, 6], in which ANN and PLS are used separately, our neural network architecture makes it possible to effectively consider complex dependences in spectral signals, which improves the results. The use of modern optimizers AdamW and RAdam, according to formulas (14)–(19), allowed us to stabilize training and achieve rapid convergence. This is explained by the adaptability of the optimizers to the uneven distribution of gradients, which is typical for spectroscopic data. All this confirms the effectiveness of the selected solutions. A series of numerical experiments conducted to optimize the model configuration made it possible to choose the best one. The architecture with two convolutional layers (32 and 64 filters), one LSTM layer (64 memory units), and ReLU activation function showed effective results. This has made it possible to achieve a prediction accuracy $R^2=0.98$ at RMSE<5 %, which is significantly superior to PCA, PLS, which demonstrated $R^2<0.85$.

The software implementation, implemented in the Python environment, allowed us to compare the performance of different configurations. The optimized configuration of the model, which was trained for 50 epochs, was selected. Dynamic adjustment of the learning rate (ReduceLROnPlateau) allowed us to reduce the training time by 25 %, while maintaining high accuracy. The main advantage of the proposed architecture over PCA and PLS used in [1] is the ability to take into account nonlinear dependences and adapt to complex conditions.

The results of testing the model on real spectra showed that the proposed model provides correct feature extraction even at low signal-to-noise ratio (<10). The model demonstrates high efficiency in gas identification: accuracy – 95.6 %, Precision – 94.8 %, Recall – 96.2 %, F1-score – 95.5 %, average error – ±0.42 nm. This confirms its ability to accurately recognize spectral features. It is also shown that the integration of neural networks will open up the possibility of automating the optimization of Fabry-Perot interferometry parameters. The analysis of gas mixtures confirmed the model's stability to changes in experimental conditions: the maximum decrease in accuracy did not exceed 3 %, while in conventional methods this figure reached 15 %. These advantages are obtained due to the fact that the model was trained on synthetic data that simulate a wide range of experimental situations. Therefore, every aspect of our results is based on a combination of architecture optimization, high-quality data preparation, and taking into account the specificity of spectroscopic analysis.

The uniqueness and advantages of the work are the proposed solutions that are provided by combining convolutional and recurrent neural networks, as well as the use of modern optimizers. The creation of synthetic training data increases the resistance to noise and variability of real conditions, and automatic normalization of spectra reduces the impact of experimental changes. Software implementation on TensorFlow/Keras provides scalability for working with big data.

The main limitation of our study is the focus only on the CNN+LSTM architecture without analyzing alternative models, such as transformers. There is also limited availability of spectral data, which may affect the generalization ability of the model, and testing under certain conditions, which does not guarantee equally high efficiency in strong noise or non-standard situations.

The main disadvantages are the dependence of the model on the quality of the training data and the high computational cost. In addition, the complexity of the architecture requires significant resources for training, which may complicate its implementation in environments with limited computing capabilities. Another drawback is the limited robustness to significant changes in the external experimental conditions (e.g., changing the spectrometer type or significantly changing the imaging conditions).

Further research may focus on improving synthetic data generation, using transfer learning to adapt the model to new tasks, optimizing the architecture to reduce computational costs, integrating cloud computing for real-time big data processing, and developing interpretive tools to increase model transparency. Prospects also include exploring adaptive learning mechanisms and using hybrid models to improve accuracy and reduce the need for big data.

7. Conclusions

1. An algorithm for automatic normalization of spectra has been developed, which includes minimax normalization, baseline correction, and noise removal using the Savitsky-Goley filter. An algorithm for generating synthetic spectra (as an option) was additionally proposed, which models experimental variations, including intensity changes, noise effects, and baseline shift. As a result, the use of synthetic data increased the accuracy of analysis by 12–15 % by reducing the influence of the variability of the experimental parameters. The algorithm demonstrates high resistance to variations in the baseline and noise levels. In particular, even with significant changes in external conditions, its accuracy remained at the level of 93–99 %. This indicates the ability of the algorithm to maintain high efficiency under real conditions, where accuracy losses do not exceed 2–3 %.

2. The proposed CNN+LSTM architecture effectively combines two technologies: CNN for extracting local features of the spectrum, such as peaks and gradients, and LSTM for analyzing global patterns in spectral signals. The use of modern optimizers AdamW and RAdam ensured stable learning and fast model convergence.

3. The software implementation of the model was performed in the Python environment (TensorFlow/Keras), which provided the possibility of scalable and efficient learning. Testing over 10 neural network configurations allowed us to find the optimal balance between performance and accuracy. The basic network configuration demonstrated fast learning (50 epochs) and stability under conditions of changing experimental parameters. Comparison with conventional PCA and PLS methods showed the advantage of the neural network in prediction accuracy: $R^2=0.98$ versus $R^2=0.85$ (PLS) and $R^2=0.80$ (PCA).

4. Experimental verification of the model on real spectral data has confirmed its effectiveness. The model successfully works with spectra with a signal-to-noise ratio <10, providing a feature extraction accuracy 30 % higher than PCA and PLS. For the tasks of gas identification in plasma, the accuracy of the model reached 95.6 %, and the average error in determining

wavelengths does not exceed ±0.42 nm. Testing on the task of analyzing the shift of spectral maxima of Fabry-Perot interferometry showed that the neural network accurately reflects the main patterns of dependences, demonstrating high consistency with experimental data.

Acknowledgments

The authors express their sincere gratitude to Doctor of Technical Sciences, Professor V. Polishchuk, and Doctor of Physical and Mathematical Sciences, Professor O. Shuaibov, for their assistance, valuable advice, and expert support, which significantly contributed to the successful completion of this study.

Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study, as well as the results reported in this paper.

Funding

The study was conducted without financial support.

Data availability

All data are available, either in numerical or graphical form, in the main text of the manuscript.

Use of artificial intelligence

The authors used artificial intelligence technologies within acceptable limits to provide their own verified data, which is described in the research methodology section.

References

1. Khayatzadeh Mahani, M., Chaloosi, M., Ghanadi Maragheh, M., Khanchi, A. R., Afzali, D. (2007). Comparison of Artificial Neural Networks with Partial Least Squares Regression for Simultaneous Determinations by ICP-AES. Chinese Journal of Chemistry, 25 (11), 1658–1662. https://doi.org/10.1002/cjoc.200790306
2. Liu, X., An, H., Cai, W., Shao, X. (2024). Deep learning in spectral analysis: Modeling and imaging. TrAC Trends in Analytical Chemistry, 172, 117612. https://doi.org/10.1016/j.trac.2024.117612
3. Primrose, M., Giblin, J., Smith, C., Anguita, M., Weedon, G. (2022). One dimensional convolutional neural networks for spectral analysis. Algorithms, Technologies, and Applications for Multispectral and Hyperspectral Imaging XXVIII, 12. https://doi.org/10.1117/12.2618487
4. Schuetzke, J., Szymanski, N. J., Reischl, M. (2023). Validating neural networks for spectroscopic classification on a universal synthetic dataset. Npj Computational Materials, 9 (1). https://doi.org/10.1038/s41524-023-01055-y
5. Liu, J., Osadchy, M., Ashton, L., Foster, M., Solomon, C. J., Gibson, S. J. (2017). Deep convolutional neural networks for Raman spectrum recognition: a unified solution. The Analyst, 142 (21), 4067–4074. https://doi.org/10.1039/c7an01371j
6. Marini, F., Bucci, R., Magrì, A. L., Magrì, A. D. (2008). Artificial neural networks in chemometrics: History, examples and perspectives. Microchemical Journal, 88 (2), 178–185. https://doi.org/10.1016/j.microc.2007.11.008
7. Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O., Walsh, A. (2018). Machine learning for molecular and materials science. Nature, 559 (7715), 547–555. https://doi.org/10.1038/s41586-018-0337-2
8. Mishra, P., Passos, D., Marini, F., Xu, J., Amigo, J. M., Gowen, A. A. et al. (2022). Deep learning for near-infrared spectral data modelling: Hypes and benefits. TrAC Trends in Analytical Chemistry, 157, 116804. https://doi.org/10.1016/j.trac.2022.116804
9. Saravanan, N., Ganesan, M. (2024). Spectral Analysis of Cellular Neural Network: Unveiling Network Parameters and Graph Characteristics. https://doi.org/10.21203/rs.3.rs-4338706/v1
10. Madden, M. G., Ryder, A. G. (2003). Machine learning methods for quantitative analysis of Raman spectroscopy data. Opto-Ireland 2002: Optics and Photonics Technologies and Applications, 4876, 1130. https://doi.org/10.1117/12.464039
11. Randolph, T. W. (2006). Scale-based normalization of spectral data. Cancer Biomarkers, 2 (3-4), 135–144. https://doi.org/10.3233/cbm-2006-23-405
12. Wei, J., Pan, S., Gao, W., Zhao, T. (2022). A dynamic object filtering approach based on object detection and geometric constraint between frames. IET Image Processing, 16 (6), 1636–1647. https://doi.org/10.1049/ipr2.12436
13. Wiatowski, T., Bolcskei, H. (2018). A Mathematical Theory of Deep Convolutional Neural Networks for Feature Extraction. IEEE Transactions on Information Theory, 64 (3), 1845–1866. https://doi.org/10.1109/tit.2017.2776228
14. Zhang, Z. (2018). Improved Adam Optimizer for Deep Neural Networks. 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), 1–2. https://doi.org/10.1109/iwqos.2018.8624183
15. Zaheer, R., Shaziya, H. (2019). A Study of the Optimization Algorithms in Deep Learning. 2019 Third International Conference on Inventive Systems and Control (ICISC). https://doi.org/10.1109/icisc44355.2019.9036442
16. Shao, X., Bian, X., Liu, J., Zhang, M., Cai, W. (2010). Multivariate calibration methods in near infrared spectroscopic analysis. Analytical Methods, 2 (11), 1662. https://doi.org/10.1039/c0ay00421a

17. Alsmeyer, F., Marquardt, W. (2004). Automatic Generation of Peak-Shaped Models. Applied Spectroscopy, 58 (8), 986–994. https://doi.org/10.1366/0003702041655421
18. Hrytsak, R., Shuaibov, O., Minya, O., Malinina, A., Shevera, I., Bilak, Y., Homoki, Z. (2024). Conditions for pulsed gas-discharge synthesis of thin tungsten oxide films from a plasma mixture of air with tungsten vapors. Physics and Chemistry of Solid State, 25 (4), 684–688. https://doi.org/10.15330/pcss.25.4.684-688
19. Shuaibov, O. K., Hrytsak, R. V., Minya, O. I., Malinina, A. A., Bilak, Yu. Yu., Gomoki, Z. T. (2022). Spectroscopic diagnostics of overstressed nanosecond discharge plasma between zinc electrodes in air and nitrogen. Journal of Physical Studies, 26 (2). https://doi.org/10.30970/jps.26.2501
20. Bondar, I. I., Suran, V. V., Minya, O. Y., Shuaibov, O. K., Bilak, Yu. Yu., Shevera, I. V. et al. (2023). Synthesis of Surface Structures during Laser-Stimulated Evaporation of a Copper Sulfate Solution in Distilled Water. Ukrainian Journal of Physics, 68 (2), 138. https://doi.org/10.15407/ujpe68.2.138
21. Kozubovsky, V. R., Bilak, Yu. Yu. (2022). Express Analysis of Gas Mixtures Using a Spectral Correlator Based on the Fabry-Perot Interferometer. Journal of Applied Spectroscopy, 89 (3), 495–499. https://doi.org/10.1007/s10812-022-01385-7
22. Kozubovsky, V., Bilak, Y. (2021). Phase Methods in Absorption Spectroscopy. Ukrainian Journal of Physics, 66 (8), 664. https://doi.org/10.15407/ujpe66.8.664