Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org

Volume 25, Issues 3, pp. 1466–1477, DOI 10.12694/scpe.v25i3.2674

ENHANCING IOT SECURITY IN RUSSIAN LANGUAGE TEACHING: A IMPROVED BPNN AND BLOCKCHAIN-BASED APPROACH FOR PRIVACY AND ACCESS CONTROL

QI JIA*

Abstract. Russian language instruction emerges as a pivotal course in tertiary education, necessitating novel approaches to maintain instructional quality and efficacy. This study introduces a novel approach to Russian language teaching that combines the robustness of Machine Learning with the security framework of Blockchain technology and is tailored to the unique needs of the Internet of Things (IoT) environment. At its core, the study creates an advanced back-propagation deep neural network enriched with a deep noise-reducing auto-encoder and a support vector machine to improve privacy and access control in IoT-based educational platforms. The proposed model employs a polynomial kernel function and a one-error penalty factor in a single hidden layer, resulting in a system that is not only efficient in handling small-scale data samples but also adept at processing larger data volumes, a common scenario in IoT settings. This design effectively overcomes the problems of overfitting and slow convergence that are common in traditional models. Furthermore, the incorporation of blockchain technology ensures a decentralized and secure data handling framework, reinforcing the privacy and access control aspects that are critical in the digital education domain. The combination of these technologies yields a more rational, scientifically based evaluation system, propelling the standardization and enhancement of Russian language instruction forward. This method not only improves language teaching quality, but it also paves the way for more secure, scalable, and efficient IoT applications in educational settings.

Key words: Teaching quality evaluation; Back propagation; Neural networks; Noise reduction; Support vector machines

1. Introduction. A major attempt to raise the calibre of instruction and teaching is teaching assessment. The assessment results provide feedback on the quality of the teachers' instruction and serve as a foundation for developing more effective teaching strategies. Learning outcomes are also reflected in teaching assessment, which can be used by students to modify their learning strategies and progress. It is an effective technique to support the management of education and teaching in a scientific and logical fashion, as well as to create a teaching force that is more targeted and concentrated. The variety of contemporary indicators for assessing teaching quality and the complexity of evaluation index aspects make it difficult to quantify a particular indicator in the teaching evaluation process during the teaching phase. The teaching process is characterised by a multi-factor loop, and the interdependence of teachers, teachers, and students creates a straightforward non-linear challenge for evaluating the quality of the instruction. Neurons are arranged in layers in non-linear systems called neural networks. Deep learning's robust information processing capabilities give teaching quality evaluation a contemporary instrument, significantly lowering the subjectivity of conventional teaching evaluation and enhancing its rationality. This shows the value and importance of using neural networks to create a model for assessing the quality of training with the goal to progress scientific teaching objectives and enhance standards for education and instruction.

The Internet of Things (IoT) has emerged as a critical component in the rapidly evolving landscape of digital education, revolutionizing how educational content, including language instruction, is delivered and managed. While this transformation provides unprecedented opportunities for interactive and personalized learning experiences, it also poses significant challenges in terms of data privacy, security, and access control. Russian language instruction in tertiary education, which is becoming increasingly important as Russia's global influence grows, is at the forefront of this digital shift. In this IoT-driven environment, the need to safeguard sensitive educational data and ensure the integrity of the teaching process is more pressing than ever. Our research focuses on developing an improved back-propagation deep neural network model with elements such

*School of International Education (Department of Foreign Language Teaching), Yellow River Conservancy Technical Institute, Kaifeng 475004, China (jiaqimark0107@outlook.com)

as a deep noise-reducing auto-encoder and a support vector machine that is specifically tailored for the context of Russian language instruction in IoT environments. This model aims to improve not only the effectiveness of language teaching but also the inherent security concerns associated with IoT-based educational systems. Within this framework, the integration of ML and Blockchain promises to deliver a more secure, efficient, and personalized educational experience.

Use of neural networks for teaching assessment relies on the development of a solid scientific model to evaluate educational quality [1, 2]. However, the low computing efficiency, sluggish convergence, and insufficient accuracy of current assessment models make further investigation and development of evaluation models necessary. To address the issues of overfitting and poor accuracy of existing models, this research suggests adaptive backpropagation neural networks and includes deep noise reduction autoencoders and support vector institutions to develop deep backpropagation neural networks on this basis. The research's goal is to create models for evaluating teaching quality that can handle samples from massive data sets.

The application of technology in an IoT environment for educational purposes. This novel approach addresses critical issues in digital education, specifically Russian language teaching. The use of these technologies in the classroom is a significant step forward in terms of improving both the quality of instruction and the security of the digital learning environment. The creation of an improved back-propagation deep neural network model that incorporates a deep noise-reducing auto-encoder and a support vector machine represents a significant step forward in the evaluation and improvement of language instruction quality. This model was created specifically to process and analyze the complexities of language teaching data, making it a useful tool for educational institutions.

2. Related Works. Numerous experts and academics have conducted a number of studies on the conventional teaching quality assessment system in an effort to enhance the teaching quality assurance system, fairly evaluate teaching quality in order to improve teaching standards, and advance education teaching towards scientific standardisation. In order to selectively label sample features, A system of active learning developed by Huang W combines Gaussian process and sparse Bayesian learning. The algorithm's improved performance was later confirmed [3]. Yuan Z analysed and evaluated feature selection techniques based on current automatic scoring systems, employed multiple regression techniques for score evaluation, and confirmed the effectiveness of the algorithm model through carefully controlled tests with the goal to expand the English translation scoring system [4]. Xiaolong developed a model employing evaluation indices from diverse viewpoints for assessing the effectiveness of online education programmes for colleges and universities. The experimental findings revealed that the algorithm model's training error was relatively small [5]. On the basis of the empirical modal decomposition approach and the adaptive complementary method, Sun Q developed the classroom theory teaching quality evaluation model and improved the correlation vector machine. After employing the baseline weights of the genetic algorithm algorithm network, the model can effectively assess the quality of English interpretation training using a process based on genetic algorithms [7].

To enhance the scientific rigour and applicability of teacher assessment, Lin L applied data mining techniques and machine learning methods for data analysis and joint model creation [8]. This was done to prevent subjectivity from influencing teaching evaluation and to advance the thoughtful growth of teaching evaluation. In order to incorporate artificial intelligence methods into classroom evaluation activities, Guo J suggested an integrated model including statistical modelling and integrated learning based on computer vision and intelligent voice recognition. The experimental findings demonstrated the model's superior functionality, with model accuracy as high as 0.905 [9]. To improve the efficiency of online teaching, Ding X et al. used association rule mining techniques for segmentation fusion and autocorrelation matching detection of teaching timeliness and developed an online teaching timeliness evaluation model based on intelligent learning. The simulation results show that the approach has a high level of confidence for assessing how timely online education is [10]. In order to assess the effectiveness of evaluating ideological and political education, Wang Y et al. employed machine learning and artificial intelligence to develop a fuzzy hierarchical analytic model of the quality of ideological and political teaching. The model uses a three-layer structure to establish a model network structure for data administration, modification, and management of the model assessment. A database for real-time updating was also built. The outcomes of the simulation experiment show that the research model meets the criteria for assessing the efficacy of ideological and political training in universities and other institutions [11]. To address

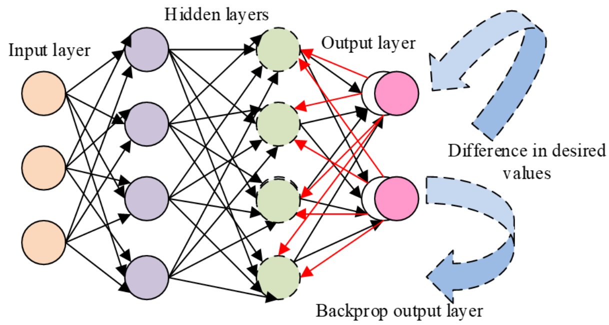


Fig. 3.1: Functional diagram of the back propagation method

the flaws in the taekwondo teaching model used in colleges and universities and to enhance the teaching effect of taekwondo, Liang H developed a taekwondo teaching effect evaluation model based on the intelligent algorithm of human feature recognition using support vector institutions. The performance of the model was confirmed using controlled trials and quantitative statistical techniques, and the concept has some practical applications in classroom education [12].

The aforementioned research on teaching quality evaluation models demonstrates that there are still some gaps in the current findings, and it is relatively uncommon to see teaching quality evaluation models built by combining deep neural networks to solve the issues of fuzzy model index weights, excessive randomness, and easy over-fitting of models, which have significant ramifications for handling large-scale data set samples.

3. Deep Neural Network Evaluation Model Construction Based on Improved BP.

3.1. Construction of a BPNN Evaluation Model Incorporating Adaptive Learning Rate and Momentum Terms. This study addresses the shortcomings of existing models and methods for processing evaluation datasets, with such improvements enhancing the gradient descent method of back propagation neural networks (BPNN) and speeding up the convergence of the model. This study also adds support for vector institutions and deep noise reduction autoencoders to the adaptive BP neural network, a change that can help to handle large evaluation sample data.

The basic processing units for algorithm learning are the neurons, which are the building blocks of artificial neural networks. The function of BPNN's back propagation approach is depicted in Figure 3.1 [13, 14]. Back propagation neural networks are more fundamental neural networks that use forward propagation for output results and back propagation for error propagation. Figure 3.1 illustrates the process of input, processing, computation and output of data in forward propagation. When the error is back-propagated, the error layer determines the discrepancy between the target's desired value and the output's actual value. The error value will adjust the neuron weights and thresholds of each layer until the error reaches the required end of the algorithm. The error is transported forward from the output layer(OL) through the HL in an inverse forward propagation way.

The study builds a three-layer BPNN using forward propagation learning to initialise the network with the number of god will elements as n, p, and q in each layers, respectively. Equation 3.1 shows the input value.

$$h_{ij}(k) = \sum_{i=1}^{n} w_{ij} x_i(k) - b_j \tag{3.1}$$

In equation 3.1, k is a sample chosen at random, $x(k)$ is the input vector, W_{ij} is the network's connection weight, and b_j is the threshold value chosen at random from the range $(-0.5, 0.5)$. Equation 3.2 displays each neuron's output value from the HL, h_o..

$$h_o(k) = f(h_{ij}(k)) \tag{3.2}$$

In accordance with 3.3, the method for determining each neuron's input value y_i in the OL based on the HL's output value, the connection weights, and the OL's threshold value is shown.

$$y_{i_t}(k) = \sum_{j=1}^{p} w_{jt} h_{o,j}(k) - b_t \tag{3.3}$$

W_{jt} is the connection weight and b_t is the threshold value in equation (3). Similar to how the input value y_i of the neuron is used to determine the output value y_o , as shown in equation 3.4.

$$y_{o,t}(k) = f(y_{i,t}(k)) \tag{3.4}$$

The target accuracy of the network is set to ϵ during the backpropagation phase of the BPNN, commonly known as error backpropagation. When the accuracy is less than the set accuracy, the bias derivatives of the neurons in the OL are calculated in equation 3.5.

$$\delta_t(k) = y_o(k)(1 - y_o(k))(t(k) - y_o(k)) \tag{3.5}$$

$\delta_t(k)$ stands for the partial derivative in equation 3.5. Equation 3.6 illustrates how the connection weights AW_{jt} and the threshold b_j AA between the HL and the OL are corrected using the derived partial derivatives and the neuron outputs of the HL. In equation 3.6, N and $N+1$ represent before and after correction respectively, and μ represents the learning step.

$$\begin{cases} w_{jt}^{(N+1)}(k) = w_{jt}^{N}(k) + \mu\delta_t(k)h_{o,j}(k) \\ b_t^{(N+1)}(k) = b_t^{N}(k) + \mu\delta_t(k) \end{cases} \tag{3.6}$$

In a similar manner, the HL neuron's partial derivative $\delta_h(k)$ is computed, as shown in equation 3.7.

$$\delta_h(k) = \left[\sum_{t=1}^{q} \delta_t(k) w_{jt}\right] h_{o,j}(k)(1 - h_{o,j}(k)) \tag{3.7}$$

The connection weights W_{ij}, b_j between the input and HLs are corrected, and the procedure is shown in equation 3.8.

$$\begin{cases} w_{ij}^{(N+1)}(k) & = w_{ij}^{N}(k) + \mu\delta_h(k)x_i(k) \\ b_j^{(N+1)}(k) & = b_j^{N}(k) + \mu\delta_j(k) \end{cases} \tag{3.8}$$

Finally, determine whether the global error meets the required precision, if so, the algorithm learning ends; otherwise, samples are chosen to recalculate the input and output values of the HL neurons until the error meets the requirements or the algorithm iteration ends. The global error calculation is illustrated in equation 3.9.

$$E = \frac{1}{2n} \sum_{k=1}^{n} \sum_{t=1}^{q} (t_t(k) - y_t(k))^2 \tag{3.9}$$

The BPNN takes a long time to train or even fails to converge well, may fall into local minima during learning, uses gradient descent to make the error converge very slowly, and the training results are unstable. To address these problems, the model enhances the BPNN by introducing adjustable learning rate and momentum components. The number of neurons n and q in the input and OLs are determined according to the input sample dimension and the output result dimension, and the number of neurons p in the HL is determined according to the empirical equation 3.10. in equation 3.10 is a constant between $[1, 10]$.

$$p = \sqrt{n+m} + a \tag{3.10}$$

The Adaptive Gradient (AdaGrad) method's learning rate dynamically adapts in response to network fault[15-16]. Equation 3.11 illustrates the process of adaptive learning rate change. In equation (11), $\mu(0)$ represents the starting learning rate, and in this investigation, β and γ have the values 1.05 and 0.7, respectively.

$$\mu(n) = \begin{cases} \beta\mu(n-1) & \text{if } E(n) < E(n-1) \quad \text{and} \quad 1 < \beta < 1.5 \\ \gamma\mu(n-1) & \text{if } E(n) > E(n-1) \quad \text{and} \quad 0.5 < \gamma < 1 \\ \mu(n-1) & \text{otherwise} \end{cases} \tag{3.11}$$

Equation 3.12 illustrates the inclusion of the momentum factor in the adaptive learning rate approach, which serves as a dampener in the process of the error back propagation correction weight. In equation 3.12, α stands for the momentum term, w for weight, and w for moment.

$$\Delta w(n) = -\mu \sum_{t=0}^{n} \alpha^{(n-t)} \frac{\partial E(n)}{\partial w(n)} \tag{3.12}$$

The weighting adjustment equation is shown in equation 3.13, where the learning rate is represented by μ and the error is represented by $E(n)$.

$$w(n+1) = w(n) - \mu(n) \sum_{t=0}^{n} \alpha^{(n-t)} \frac{\partial E(n)}{\partial w(n)} \tag{3.13}$$

3.2. Deep Neural Network Evaluation Model Construction Based on Improved BPNN. The momentum terms as well as the adaptive learning rate in the BPNN evaluation model have certain advantages when handling small-scale datasets, but their capacity to handle complicated and high-dimensional large dataset samples is constrained. The paper builds a deep network model to solve this issue by layering deep noise reduction autoencoders over BP neural networks and adding Support Vector Regression (SVR) to the OL.

Artificial neural networks are deepened by deep neural networks, which have many HL. Deep noise reduction autoencoders have a stronger ability to extract essential features than the original autoencoders because they consist of many autoencoders that add noise to the data set to prevent overfitting during training [17, 18].

As seen in Figure 3.2, the feature of zeroing is mostly used for noise reduction processing of the noise contained in the input original data. First, set a particular probability to set part of the data in the original matrix x to 0 to get the residual input matrix $\widetilde{x}$ with lost data. The compressed matrix y is obtained by layer-by-layer coding, followed by layer-by-layer pass to obtain x' , error between x and x' for network parameter learning, and iteration to obtain the compressed coded y . The entire training process improved in robustness and generalizability.

To minimise the error and complete reconstructing the original input dataset, the error between the reconstructed dataset and the original dataset is then calculated using an error function, and the BP algorithm is used to propagate the error to the entire depth noise reduction autoencoder and modify the weights and thresholds. The cost function is the mean squared error function, whose expression is given in equation; the weights and thresholds are updated using the gradient descent method 3.14.

$$L(x,y) = \frac{1}{n} \sum_{i=1}^{n} (y_i - x_i)^2 \tag{3.14}$$

The Adam algorithm combines the two well-known techniques "Adagrad" (for sparse gradients) and "RMSPro" to solve optimisation issues involving vast amounts of data and high feature latitude. (for non-stationary data). Figure 3.3 illustrates how the Adam approach, which is computationally efficient and ideal for very noisy and sparse gradient issues, can replace the conventional random gradient descent method to update the network weights more effectively. It also functions as a deep noise reduction autoencoder. The deep noise reduction autoencoder's OL neurons are shown as dashed circles in Figure 3.3, while the true output is a classifier or predictor.

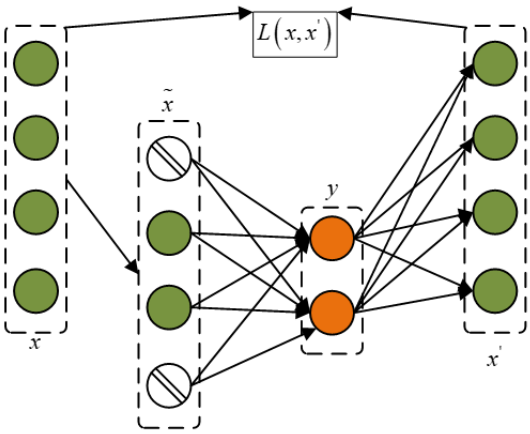


Fig. 3.2: Structure of Noise Reduction Automatic Encoder

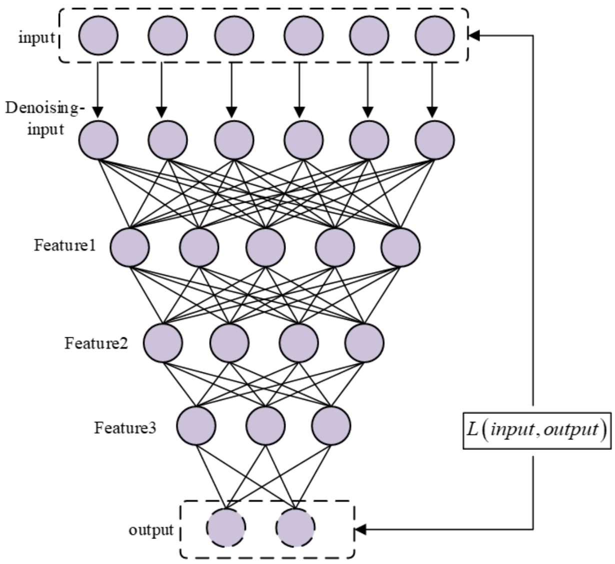


Fig. 3.3: Structure of deep automatic noise reduction encoder

Support Vector Machines (SVM) are a subclass of generalised linear classifiers that conduct supervised learning (supervised learning) binary categorization of data for tasks including regression, classification, and recognition. A linear classifier is a line that divides two groups of data in a two-dimensional plane. A plane serves as a linear classifier in three dimensions. In higher dimensions, a hyperplane is created from a linear classifier. Linear and non-linear regression are two categories for support vector analysis. In the former, complex nonlinear relationships are mapped into a high-dimensional space, where they are then realised and behave like linearized relationships in the latter [19, 20]. In response to the various indications for evaluating the quality of Russian language education and the complex non-linear connection between indicators and evaluation conclusions, the study uses support vector non-linear regression.Support vector regression serves as the predictor for the final OL in the deep noise reduction auto-coding unsupervised training layer of the final deep neural network evaluation model built using the improved BPNN, and the error between the original input dataset and the unsupervised

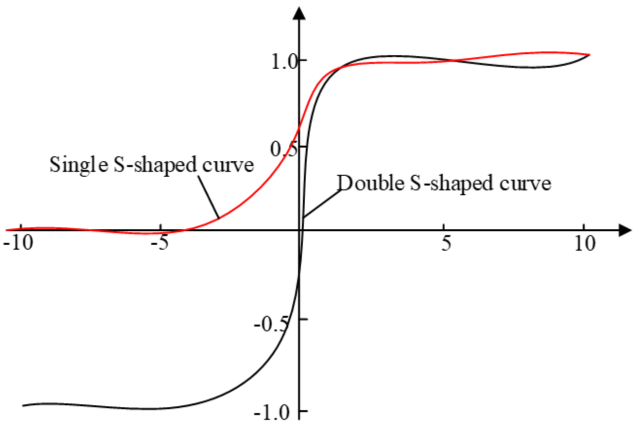


Fig. 3.4: Sigmoid function curve

training output data is minimised to obtain the feature vectors of the original input dataset. The support vector regression model's structure is depicted in

The data is often pre-processed before being fed into the model, and the study normalises the sample data to transform the data between intervals . One way to lessen the difficulty of weight adjustment is to scale back the amount of the input value change. On the other hand, Figure 4.1 depicts the activation function of the BPNN as a double S-shaped curve.

The transformation between [-1,1] and the derivative of the activation function is identical, Which is shown in Equation 3.15.

$$y = f(x) = \frac{1 - e^x}{1 + e^x} \tag{3.15}$$

The normalisation operation can speed up the convergence of the network and improve the computational efficiency. The operation process is shown in equation . In equation , x_{max} and x_{min} denote the maximum and minimum values in the data, x_i and z_i denote the data output before and after processing, and x_{mid} denotes the intermediate values of data changes.

$$\begin{cases} z_i = \frac{x_i - x_{min}}{\frac{1}{2}(x_{max} - x_{min})} \\ x_{mid} = \frac{x_{max} + x_{min}}{2} \end{cases} \tag{3.16}$$

4. Performance Testing of Improved BP Deep Neural Network Evaluation Models.

4.1. Test Experimental Protocol Design and Model Parameter Analysis. A test experiment was created to confirm the effectiveness of the built model. The experiment identifies metrics for assessing the effectiveness of Russian language instruction from two perspectives—student evaluation and teaching supervision groups—as well as from two dimensions—preparation before instruction and during instruction. These metrics include teaching attitude, teaching content, teaching methods, and answering questions after class. They also include professional quality, teaching ability, preparation before instruction, and the energy of the classroom environment. The dataset originates from a university academic system's dataset on the evaluation of Russian language courses and contains 3684 examples of data. Student evaluations are used as model input values and the evaluations of the teaching supervisory team are used as the target expectation values of the algorithm model. Finally, all data are normalised to increase the algorithm's calculation efficiency.

The amount of neurons in the input layer was set to 30, and the amount of neurons in the output layer to 1, the growth ratio of the adaptive learning rate to 1, the decline ratio to 0, and the momentum term to 0.65

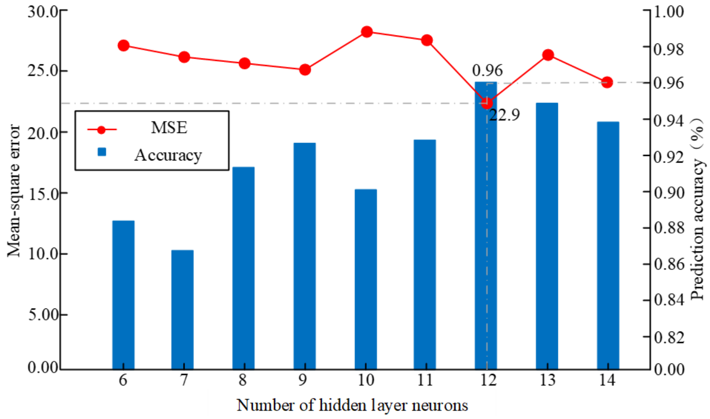


Fig. 4.1: Effect of the number of neurons on mean square error and accuracy

in order to compute the number of neurons in the HL. The number of neurons in the HL was determined by applying Equation 3.10 and utilising the mean-square error (MSE) and prediction accuracy as the assessment indices. The optimum amount of neurons for the HL was determined using Mean Square Error (MSE) and prediction accuracy. The training outcomes are shown in Figure 4.1. Figure 4.1 demonstrates that the mean squared error is at a minimum of 22.9 and the prediction accuracy is at a maximum of 0.96 when there are 12 neurons in the HL. The minimal prediction accuracy is only approximately 86%, and the mean squared error does not change significantly when the number of neurons changes, but the accuracy value fluctuates more. The evaluation model's accuracy is taken into account for determining the HL's number of neurons, which is set at 12.

2, 3, 4, and 5 HLs, together with 12 HL neurons, were chosen as the parameters. The unsupervised training was done using Adam's technique, and the evaluation index was the difference in error between the reconstructed data feature vector and the original data set. Figure 7 displays the training outcomes after 5000 iterations. Figure 4.2 illustrates how the error value curves all exhibit a declining trend as the number of iterations rises. The model with two HLs exhibits the highest decline in error value, with a 68.% drop from the start of the iteration. When there are just two HLs, the algorithm model's training result is perfect; nevertheless, with the same number of repetitions, the error value climbs steadily as the number of HLs rises.

The penalization coefficient and the kind of kernel function are the two primary factors influencing support vector regression in the supervised prediction output process. Model complexity and empirical riskiness are both impacted by the penalty coefficient, and modifying these two factors enhances the algorithm's overall performance. To calculate the Mean Absolute Percentage Error (MAPE) between the predicted evaluation result value and the actual evaluation value, the penalty coefficients are taken to be in the range of 1 to 9, and the kernel functions are taken into consideration to be Liner, Poly, radial basis function, and Sigmoid function. Figure 4.3 displays the model training outcomes and the evaluation index, the MAPE. In accordance to Figure 8, despite the lesser degree of error fluctuation, the excessive penalty still causes the MAPE to be excessively large. The MAPE of the support vector machine consisting of all kernel functions roughly tends to increase as the error penalty factor increases. In comparison to the other three types of functions, the polynomial function has the significantly lowest MAPE value, with a MAPE value of only 0.0506 when the penalty coefficient is 1. Support vector regression uses the polynomial function as its kernel function.

4.2. Quality Analysis of Model Training Results. With two HLs, 12 neurons each, and an error penalty value of 1, the adaptive learning rate was set to 1.1 for growth ratio, 0.8 for decline ratio, and 0.65 for momentum term. The kernel function for the training of the model was decided to be a polynomial function. To compare the MAPE, MSE, Root Mean Square Error (RMSE), and Symmetric Mean Absolute Percentage Error (SMAPE) of the various algorithms, the BP deep neural network developed in this study was first compared

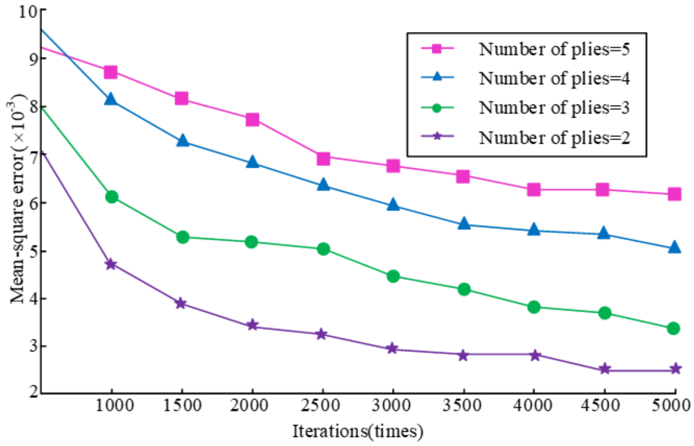


Fig. 4.2: Effect of the number of HLs on mean square error

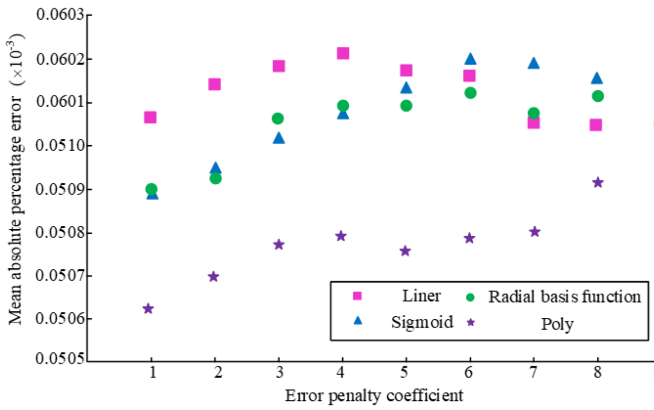
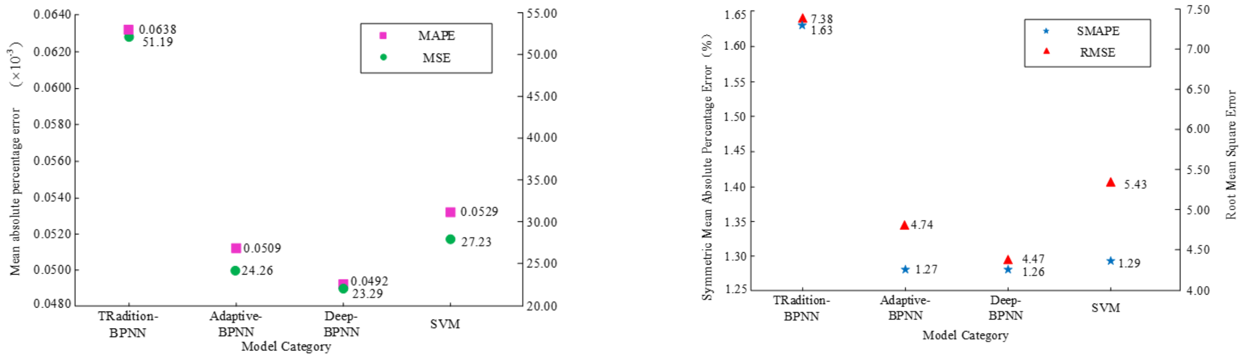


Fig. 4.3: Effect of error penalty coefficient on MAPE of different kernel functions

with the adaptive BPNN, traditional BPNN, and support vector machine algorithms. Figure 8 displays the training outcomes and the RMSE. Figure 4.4 shows that the BP deep neural network developed in this study had the lowest values for all four metrics, with MAPE of 0.0492, MSE of 23.29, SMAPE of 1.26, and RMSE of 4.47. The typical BP neural network had higher error values for all four metrics. In particular, the MAPE and MSE measures were 25.34 and 111 percentage points higher than the BP deep neural network's, making them inappropriate for use as direct assessment models in comparison. The SMAPE values of the adaptive BPNN were the ones that were closest to those of the BP deep neural network. Although they performed slightly worse in terms of the magnitude of the other three errors, overall performance was not significantly different, proving that the construction of adaptable BP neural networks was correct and highlighting the need for further advancements in adaptable BP neural networks. When the algorithm learning was finished, the comparison of training time and accuracy is continued, and the results are displayed in Figure 4.5. Figure 4.5 demonstrates that the enhanced adaptive BP neural network and the deep neural network have much higher accuracy values, up to 5 percentage points higher than the traditional BP neural network. However, the accuracy rates of the four networks are not significantly different. The adaptive BP neural network, however, took the shortest amount of time to train—only 1.07s—a difference of 10.63 seconds from the traditional BPNN and 2.9 seconds from the deep BP neural network. This shows that the adaptive BP neural network handles the concerns with



(a) MAPE, MSE

(b) SMAPE, RMSE

Fig. 4.4: Comparison of network performance of different algorithms

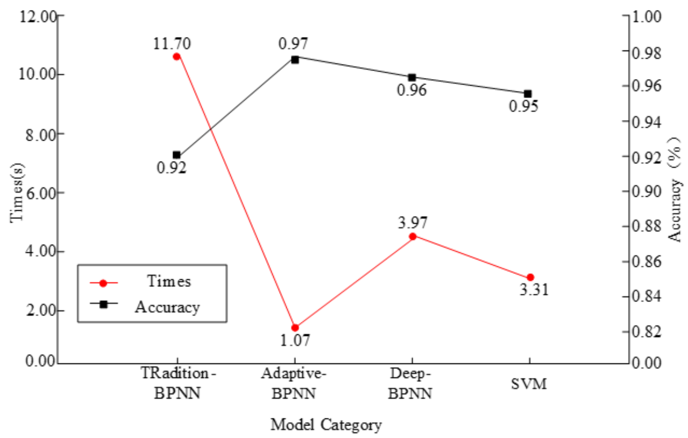


Fig. 4.5: Network training time and accuracy of different algorithms

delayed convergence and slipping into local minima that the classic neural network experiences and is more suited to handling tiny data sample sets. The deep BPNN exhibits some improvements in error values, but these advantages are not significant because the deep BPNN's structure is complex and the number of HLs grows, which lengthens processing time for small sample sets. The performance of the BP deep neural network was then compared when it was used with various optimisation techniques, such as BPNN-Gradient Descent, BPNN-Momentum, and BPNN-RMSProp. The training results are displayed in Figure 4.6s. The BP deep neural network and the BPNN-RMSProp algorithms' mean squared error values reduced the quickest and had the sharpest curve trend below 1000 iterations, as can be seen in Figure 4.6. As the number of iterations increases, the error curve flattens out and the error values do not decrease significantly, even though the BPNN created using Adam's optimisation algorithm in this study had the best results in terms of reconstructing the input data at the end of unsupervised learning training and had the lowest error values at the end of the iterations. The error values of the Gradient Descent algorithm and Momentum algorithm also showed a decreasing trend, but the error values were larger and the algorithm's overall training performance was not better. However, the error curves of the Gradient Descent and Momentum algorithms also show a decreasing trend. Finally, the sample data were normalised on the large-scale dataset to highlight the benefits of BP deep

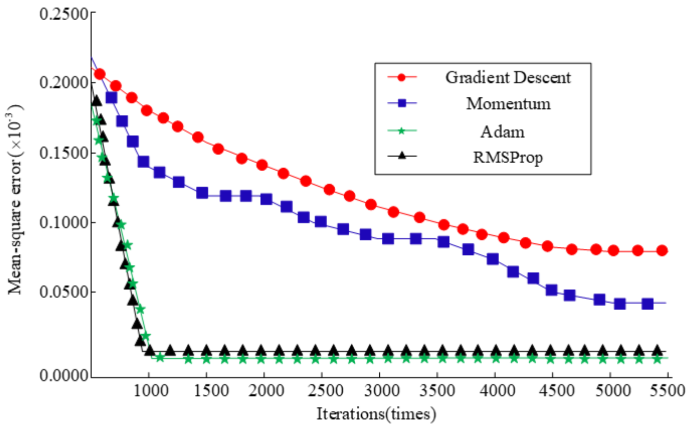


Fig. 4.6: Performance Comparison of Different Optimization Algorithms BPNN

Table 4.1: Performance Comparison of Large Datasets

Model	MAPE	MSE	SMAPE	RMSE	Times
Adaptive-BPNN	0.2453	68.9	6.890	9.087	45.98
Deep-BPNN	0.0876	28.6	3.002	7.930	79.24
TRadition-BPNN	1.6274	106.6	8.236	11.231	123.69
SVM	0.8952	89.7	7.263	8.563	86.66

neural networks on large-scale datasets. The final training results are displayed in Table 1 after comparing the MAPE, MSE, SMAPE, and RMSE values and training duration of the four networks. Table 4.1 demonstrates how Deep-BPNN outperforms Adaptive-BPNN in terms of error performance measures when processing massive datasets. These metrics are all significantly lower for Deep-BPNN. The training duration was 79.24 seconds, but even though there were more HLs and total HL neurons, the training time was still within a reasonable range.

5. Conclusion. This study enhanced the conventional BP neural network by including a support and a deep noise reduction autoencoder vector mechanism to the adaptive learning BPNN to create a deep neural network to meet the challenging nonlinear problem of evaluating the quality of teaching Russian. The results of the model performance test indicate that 12 neurons, with a mean squared error of 22.9 and a prediction accuracy of 0.96, are the ideal number for a single HL. When there are two HLs, the error curve of the built-in deep neural network model shrinks the quickest, reaching a maximum reduction of 68.3%. The error penalty factor was adjusted to 1 using the polynomial function, which is best for enhancing the algorithm's overall performance. The MAPE value at the end of the model training was only 0.0506. With a MAPE of 0.0492, an MSE of 23.29, a SMAPE of 1.26, and an RMSE of 4.47, the BP deep neural network outperformed the adaptive BPNN, regular BPNN, and support vector machine algorithms in terms of error values and accuracy magnitudes. The adaptive BP neural network is better suited for processing small-scale data sample sets because its error value is marginally larger than that of the BP deep neural network, but its training time is shorter—only 1.07s, 10.63s less than the traditional BPNN and 2.9s less than the deep BP neural network. With the lowest error value at the end of the iterations and the highest performance at reconstructing the input data throughout unsupervised learning training, the deep BPNN built utilising Adam's optimisation technique clearly has an edge when working with large-scale data sets. Further study is still required to determine the effectiveness and duration of training for the built-in deep neural network model.

REFERENCES

[1] Rong, Z. & Gang, Z. An artificial intelligence data mining technology based evaluation model of education on political and ideological strategy of students. *Journal Of Intelligent & Fuzzy Systems: Applications In Engineering And Technology.* **40**, 3669-3680 (2021)
[2] Liu, G. & Zhuang, H. Evaluation model of multimedia-aided teaching effect of physical education course based on random forest algorithm. *Journal Of Intelligent Systems.* **31**, 555-567 (2022)
[3] Huang, W. Simulation of English teaching quality evaluation model based on gaussian process machine learning. *Journal Of Intelligent And Fuzzy Systems.* **40**, 2373-2383 (2021)
[4] Yuan, Z. Interactive intelligent teaching and automatic composition scoring system based on linear regression machine learning algorithm. *Journal Of Intelligent & Fuzzy Systems: Applications In Engineering And Technology.* **40**, 2069-2081 (2021)
[5] Wen, X. Research on the teaching quality evaluation model of distance education in colleges based on analytic hierarchy process. *International Journal Of Continuing Engineering Education And Life-Long Learning.* **32**, 796-810 (2022)
[6] Sun, Q. Evaluation model of classroom teaching quality based on improved RVM algorithm and knowledge recommendation. *Journal Of Intelligent And Fuzzy Systems.* **40**, 2457-2467 (2021)
[7] Lu, C., He, B. & Zhang, R. Evaluation of English interpretation teaching quality based on GA optimized RBF neural network. *Journal Of Intelligent And Fuzzy Systems.* **40**, 3185-3192 (2021)
[8] Lin, L. Smart teaching evaluation model using weighted naive bayes algorithm. *Journal Of Intelligent & Fuzzy Systems: Applications In Engineering And Technology.* **40**, 2791-2801 (2021)
[9] Guo, J., Bai, L., Yu, Z., Zhao, Z. & Wan, B. An AI-Application-Oriented In-Class Teaching Evaluation Model by Using Statistical Modeling and Ensemble Learning. *Sensors.* **21**, 241-251 (2021)
[10] Ding, X., Salam, Z. & Lv, W. Research on Timeliness Evaluation Model of Online Teaching Based on Intelligent Learning. *International Journal Of Continuing Engineering Education And Life-Long Learning.* **31**, 263-275 (2021)
[11] Wang Y. Ideological and political teaching model using fuzzy analytic hierarchy process based on machine learning and artificial intelligence. *Journal Of Intelligent & Fuzzy Systems: Applications In Engineering And Technology.* **40**, 3571-3583 (2021)
[12] Liang, H. Role of artificial intelligence algorithm for taekwondo teaching effect evaluation model[J]. *Journal Of Intelligent And Fuzzy Systems.* **40**, 3239-3250 (2021)
[13] Zeng, Q., Jiang, H., Liu, Q., Li, G. & Ning, Z. Design of a high-temperature grease by BP neural network and its preparation and high-temperature performance studies. *Industrial Lubrication And Tribology.* **74**, 564-571 (2022)
[14] Zhu, D., Cheng, C., Zhai, W., Li, Y. & Chen, B. Multiscale Spatial Polygonal Object Granularity Factor Matching Method Based on BPNN. *International Journal Of Geo-Information.* **10**, 75-91 (2021)
[15] Xue, Y., Wang, Y. & Liang, J. self-adaptive gradient descent search algorithm for fully-connected neural networks. *Neurocomputing.* **478**, 70-80 (2022)
[16] Bharathi, L. & Chandrabose, S. Machine Learning-Based Malware Software Detection Based on Adaptive Gradient Support Vector Regression. *International Journal Of Safety And Security Engineering: An Interdisciplinary Journal For Research And Applications.* **12**, 39-45 (2022)
[17] Chen, F., Liu, L., Tang, B. & Zhang, F. novel fusion approach of deep convolution neural network with auto-encoder and its application in planetary gearbox fault diagnosis[J]. *Journal Of Risk And Reliability.* **235**, 3-16 (2021)
[18] Wu, H., Liu, S., Cheng, C., Cao, S., Cui, Y. & Zhang, D. Multiscale variational autoencoder aided convolutional neural network for pose estimation of tunneling machine using a single monocular image. *IEEE Transactions On Industrial Informatics.* **18**, 5161-5170 (2021)
[19] Sweeti Attentional load classification in multiple object tracking task using optimized support vector machine classifier: a step towards cognitive brain–computer interface. *Journal Of Medical Engineering & Technology.* **46**, 69-77 (2022)
[20] Sun, F. & Shi, G. Study on the application of big data techniques for the third-party logistics using novel support vector machine algorithm. *Journal Of Enterprise Information Management.* **35**, 1168-1184 (2022)

Edited by: Kumar Abhishek
Special issue on: Machine Learning and Block-chain based Solution for Privacy and Access Control in IoT
Received: Oct 7, 2023
Accepted: Dec 19, 2023