Journal of Artificial Intelligence, Virtual Reality, and Human-Centered Computing ISSN: 3069-2318

PRIME OPEN ACCESS

Volume 2, Issue 2

Research Article

Date of Submission: 09 Apr, 2026
Date of Acceptance: 15 May, 2026
Date of Publication: 05 Jun, 2026

Optimized Neural Network Architecture for High-Accuracy Data Classification

Mohd Nadeem*

Department of Computer Science and Engineering, Shri Ramswaroop Memorial University, India

***Corresponding Author:** Mohd Nadeem, Department of Computer Science and Engineering, Shri Ramswaroop Memorial University, India.

Citation: Nadeem, M. (2026). Optimized Neural Network Architecture for High-Accuracy Data Classification. *J AI VR Hum Comput, 2*(2), 01-11.

Abstract
Accurate data classification plays a critical role in modern intelligent systems, particularly in areas such as healthcare, finance, cybersecurity, and software engineering, where reliable decision-making depends on precise data analysis. Traditional machine learning algorithms often encounter challenges when dealing with complex, high dimensional, and large-scale datasets, resulting in reduced predictive accuracy and limited adaptability. Neural networks have demonstrated strong potential in addressing these challenges; however, their performance heavily depends on architectural design, parameter configuration, and training strategies. Therefore, developing optimized neural network architecture is essential to improve classification accuracy and computational efficiency. This study proposes an optimized neural network architecture designed to enhance high-accuracy data classification across diverse datasets. The proposed framework focuses on systematic optimization of the neural network structure, including the selection of appropriate hidden layers, neuron configurations, activation functions, and learning parameters. In addition, preprocessing techniques such as data normalization, feature scaling, and dimensionality handling are incorporated to improve the learning capability of the model. Regularization mechanisms and dropout techniques are also applied to reduce overfitting and improve generalization performance. The proposed architecture is evaluated using benchmark datasets commonly used in classification research. The performance of the optimized model is compared with conventional machine learning approaches and baseline neural network models. Evaluation metrics such as accuracy, precision, recall, F1-score, and receiver operating characteristic–area under the curve (ROC-AUC) are employed to assess classification effectiveness and robustness. Experimental results demonstrate that the optimized neural network architecture significantly improves classification performance and learning stability. The model achieves higher predictive accuracy and better generalization compared to traditional models, while maintaining efficient computational performance. The findings indicate that architectural optimization and proper training strategies play a vital role in improving neural network-based classification systems. The proposed framework provides an effective and scalable approach for high-accuracy data classification. It can be applied to various real-world applications involving complex datasets, and it offers a promising direction for future research in intelligent data-driven decision-making systems.

Keywords: Neural Network Architecture, Data Classification, Machine Learning, Deep Learning, Hyperparameter Optimization, Feature Engineering, Classification Accuracy, Artificial Intelligence

Introduction
The rapid growth of digital data across various domains has created an increasing demand for accurate and efficient data classification techniques. Data classification plays a crucial role in many real-world applications such as healthcare diagnosis, financial fraud detection, cybersecurity threat identification, software defect prediction, and intelligent decision-support systems [1]. With the expansion of big data environments, traditional machine learning algorithms often struggle to maintain high accuracy due to complex data distributions, high dimensionality, and nonlinear relationships among variables. As a result, advanced computational models capable of learning complex patterns are required to improve classification reliability and performance [2].

Artificial Neural Networks (ANNs) have emerged as powerful tools for solving complex classification problems due to their ability to model nonlinear relationships and automatically learn hierarchical data representations [3]. Neural networks are widely applied in image recognition, medical diagnostics, natural language processing, and predictive analytics. However, the performance of neural networks largely depends on their architectural design, including the number of hidden layers, neuron distribution, activation functions, learning rates, and optimization strategies. Poor architectural configuration can lead to overfitting, slow convergence, increased computational cost, and reduced classification accuracy. In recent years, researchers have focused on improving neural network performance through architectural optimization, hyperparameter tuning, and hybrid machine learning frameworks [4,5]. Despite these efforts, many existing classification models still rely on fixed neural network structures that may not be suitable for complex or heterogeneous datasets. Additionally, improper parameter selection often limits the generalization ability of neural networks, leading to suboptimal predictive performance. Therefore, designing an optimized neural network architecture capable of achieving high classification accuracy while maintaining computational efficiency remains a critical research challenge. Beyond technical performance, inaccurate data classification can lead to significant economic losses across industries [6]. Misclassification in cybersecurity systems may allow cyberattacks to go undetected, resulting in data breaches and financial damage. Similarly, errors in medical diagnosis systems can increase healthcare costs and patient risk, while inaccurate financial fraud detection can cause large economic losses for financial institutions. The increasing financial impact of data misclassification highlights the urgent need for more reliable and optimized machine learning models.

Year	Sector	Estimated Global Loss (USD)	Cause of Loss	Reference Trend
2020	Cybersecurity	$3.86 Billion (average breach cost)	Misclassification of threats	Industry cybersecurity reports
2021	Financial Fraud	$5.38 Billion	Fraud detection misclassification	Financial crime studies
2022	Healthcare Data Errors	$6.2 Billion	Diagnostic classification errors	Healthcare analytics reports
2023	Cybercrime	$8.15 Trillion	AI and system vulnerability exploitation	Global cybersecurity reports
2024	Cybercrime	$9.5 Trillion	Advanced cyberattacks and misdetection	Cybersecurity forecasts
2025	Cybercrime	$10.5 Trillion	AI-driven cyber threats	Global cyber risk studies
2026 (Projected)	Cybercrime	$11.5 Trillion	Increasing automated attacks	Security industry predictions

Table 1: Monetary Loss Due to Data Misclassification and Cyber Incidents

The table 1 illustrates the growing financial impact associated with inaccurate classification and detection systems. These losses emphasize the importance of developing highly accurate classification models capable of minimizing misclassification errors in critical systems [7-10].

Research Gap

• Although neural networks have demonstrated promising results in classification tasks, several limitations remain in existing studies:
• Many traditional neural network models use static architectures that are not optimized for complex or heterogeneous datasets.
• Existing approaches often lack systematic architectural optimization, resulting in reduced model efficiency and accuracy.
• Limited attention has been given to balancing classification accuracy with computational efficiency, particularly for large datasets.
• Many models fail to incorporate robust regularization and feature preprocessing mechanisms, which are essential for improving generalization performance.

Importance of the Study

To address these challenges, this research focuses on developing an optimized neural network architecture for high-accuracy data classification. The proposed approach aims to improve classification performance by optimizing neural network structure, training parameters, and preprocessing techniques. By enhancing predictive accuracy and reducing classification errors, the proposed framework can contribute to more reliable decision-making systems in various application domains such as healthcare, cybersecurity, finance, and intelligent data analytics. Moreover, improved classification accuracy can significantly reduce economic losses associated with misclassification and system vulnerabilities.

Related Works

In recent years, neural networks and deep learning techniques have gained significant attention for solving complex data classification problems across domains such as healthcare, finance, cybersecurity, and software engineering [11]. The ability of neural networks to automatically learn hierarchical feature representations has significantly improved classification accuracy compared with traditional machine learning algorithms. However, achieving optimal performance still depends on factors such as network architecture design, hyperparameter optimization, training strategies, and data preprocessing methods [12].

Early research in neural networks focused on basic artificial neural network (ANN) architectures such as feedforward neural networks and multilayer perceptrons (MLP) [13]. These models demonstrated the ability to capture nonlinear relationships within datasets but often suffered from issues such as overfitting, slow convergence, and limited scalability for high-dimensional data. With the advancement of deep learning, more complex architectures such as Convolutional Neural Networks (CNNs) Recurrent Neural Networks (RNNs) and transformer-based models have been developed to address complex classification tasks [14,15]. CNNs, in particular, have achieved remarkable success in image classification and pattern recognition due to their capability to automatically extract spatial features from data. Recent studies have emphasized optimizing neural network architectures to enhance classification accuracy and computational efficiency. Neural Architecture Search (NAS) has emerged as an automated technique for discovering optimal network structures by exploring a large search space of possible architectures. NAS algorithms can automatically identify suitable layer configurations and parameter settings, significantly improving model performance compared to manually designed architectures [16].

Several researchers have also explored evolutionary and metaheuristic optimization methods for neural network design. Genetic algorithms, particle swarm optimization (PSO), and differential evolution techniques have been applied to determine optimal network structures and hyperparameters. These methods can effectively explore the architecture search space and avoid local minima during training [17].

Optimization algorithms also play a crucial role in improving neural network performance. Traditional gradient-based methods such as stochastic gradient descent (SGD) and its variants, including Adam and RMSProp, have been widely used to accelerate training and improve convergence rates. Recent studies have also explored advanced optimization strategies, including quasi-Newton methods, natural gradient descent, and gradient-free optimization approaches, which can further enhance classification performance and training stability [18].

Despite these advancements, several challenges remain in designing neural network architectures that achieve high classification accuracy while maintaining computational efficiency. Many existing models rely on manually designed architectures that may not generalize well across different datasets. Additionally, architecture optimization techniques often require significant computational resources, limiting their practical applicability. Therefore, developing an optimized neural network framework that balances classification accuracy, computational efficiency, and model scalability remains an important research direction.

No.	Author / Year	Method /Model	Dataset / Application	Key Contribution	Limitation
1	LeCun et al., 2015 [19]	Deep Learning	Image & speech recognition	Introduced deep learning concepts	High computational cost
2	Krizhevsky et al., 2012 [20]	CNN (AlexNet)	ImageNet	Breakthrough in image classification	Large model size
3	He et al., 2016 [21]	ResNet	Image classification	Residual learning for deeper networks	Complex architecture
4	Iandola et al., 2016 [22]	SqueezeNet	Image classification	Smaller model with high accuracy	Limited generalization
5	Vaswani et al., 2017 [23]	Transformer	NLP classification	Attention-based architecture	High training cost
6	Sun et al., 2017 [24]	Genetic CNN	Image classification	Evolutionary architecture search	Computationally expensive
7	Zoph & Le, 2017 [25]	NAS	Image classification	Automated architecture discovery	Requires huge compute
8	Real et al., 2018 [26]	Evolutionary NAS	Image classification	Architecture optimization	Long training time
9	Tan & Le, 2019 [27]	EfficientNet	Image classification	Scalable CNN architecture	Hardware dependency
10	Bashar, 2019 [28]	Deep Neural Networks	General classification	Survey of DL architectures	Limited empirical analysis
11	Rahaman et al., 2019 [29]	Spectral Bias Study	Neural network learning	Understanding learning behavior	Theoretical focus
12	Jamous et al., 2021 [30]	PSO-based ANN	Classification tasks	Optimization of ANN structure	Parameter sensitivity
13	Alzubaidi et al., 2021 [31]	Deep learning survey	Medical imaging	Review of CNN architectures	Domain-specific focus
14	Belciug et al., 2022 [32]	Differential Evolution	Neural architecture learning	Automatic architecture optimization	Computational complexity
15	Ang et al., 2022 [33]	TLBO optimization	CNN architecture design	Automatic network design	Limited datasets
16	Abdulkadirov et al., 2023 [34]	Optimization algorithms	Neural network training	Improved convergence techniques	Complexity in tuning

17	Poyser et al., 2024 [35]	NAS survey	Image classification	Review of modern NAS methods	Limited benchmarking
18	Salehin et al., 2024 [36]	AutoML	ML model automation	Automated model building	High computational overhead
19	Latif et al., 2024 [37]	CNN optimization	Medical image classification	Architecture optimization methods	Domain limitation
20	Menéndez et al., 2024 [38]	ANN classification	Multiple datasets	ANN for classification tasks	Limited optimization techniques
21	Zhou et al., 2025 [39]	Multi-stream CNN	Image classification	Improved feature extraction	Complex architecture
22	Ji et al., 2025 [40]	DARTS-VAN	Image classification	NAS-based architecture optimization	Resource intensive

Table 2: Literature Review Comparison Table

From the reviewed literature:
- Most studies focus on specific neural architectures (CNN, RNN, Transformer) rather than generalized optimized ANN frameworks.
- Existing Neural Architecture Search techniques require high computational resources.
- Many studies focus on domain-specific datasets, limiting model generalization.
- There is a lack of balanced frameworks combining architecture optimization, feature preprocessing, and training efficiency.
- Designing an optimized neural network architecture adaptable to multiple datasets.
- Improving classification accuracy with reduced computational complexity.
- Integrating feature preprocessing, hyperparameter tuning, and architecture optimization into a unified framework.

Materials and Methods

This section describes the datasets, preprocessing procedures, neural network architecture design, optimization strategies, and evaluation metrics used in the proposed Optimized Neural Network Architecture for High-Accuracy Data Classification. The goal of the proposed framework is to improve classification accuracy by optimizing the neural network structure, training parameters, and feature preprocessing mechanisms [19].

Research Framework: The proposed research framework consists of multiple stages, including data acquisition, preprocessing, feature extraction, neural network architecture optimization, model training, and performance evaluation. The framework is designed to ensure efficient learning, improved generalization capability, and high classification accuracy across different datasets.

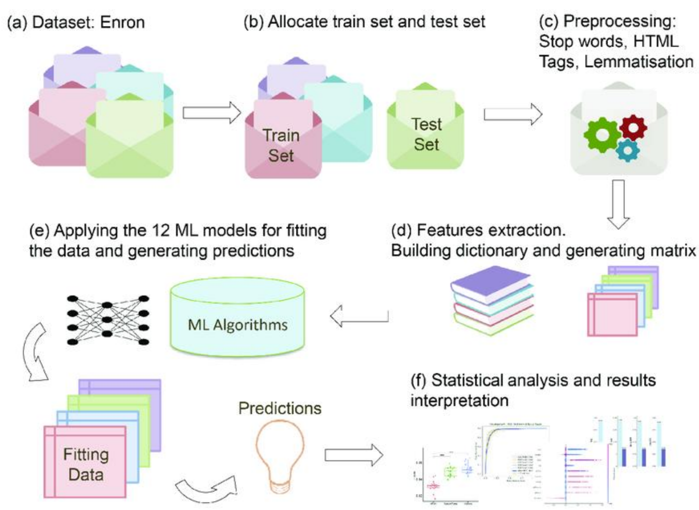

Figure 1: Research Framework

The framework operates in the following sequential stages:
Data Acquisition – Collection of benchmark datasets for classification tasks.
Data Preprocessing – Cleaning, normalization, and feature scaling to improve data quality.

Feature Engineering – Extraction and transformation of relevant features.
Optimized Neural Network Design – Development of the proposed architecture with optimized hyperparameters.

Model Training – Training the neural network using optimized learning algorithms.
Model Evaluation – Evaluating the classification performance using standard metrics.

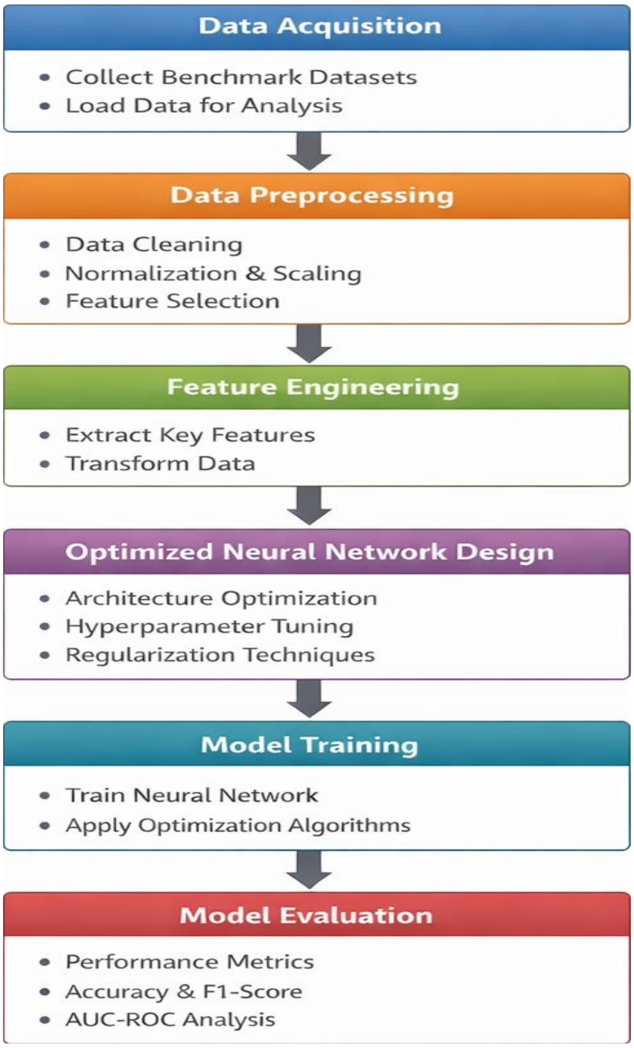

Figure 2: Proposed Flowchart

Dataset Description: To evaluate the performance of the proposed neural network architecture, benchmark datasets commonly used in classification research are utilized. These datasets contain structured and labeled data suitable for supervised learning experiments.

Dataset	Domain	Number of Instances	Number of Features	Classes
Iris Dataset	Pattern Recognition	150	4	3
Breast Cancer Wisconsin	Medical Diagnosis	569	30	2
Wine Dataset	Chemical Analysis	178	13	3
PIMA Diabetes Dataset	Healthcare	768	8	2
Heart Disease Dataset	Clinical Prediction	303	14	2

Table 3: Dataset Description

These datasets provide diverse feature distributions and classification challenges, making them suitable for evaluating the effectiveness of optimized neural network architectures.

Data Preprocessing: Data preprocessing is an essential step in machine learning and neural network-based classification models. Raw datasets often contain noise, missing values, and inconsistent feature scales that can negatively affect model performance.

The preprocessing stage includes the following operations:

• **Data Cleaning:** Incomplete records and missing values are handled using statistical imputation techniques such as mean or median substitution.

• **Feature Normalization:** Features are normalized using Min–Max scaling to ensure that all input variables lie within a consistent range.

$$X_{Norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

• Normalization improves the convergence speed of neural network training.

• **Feature Selection:** Irrelevant or redundant features are removed to reduce dimensionality and improve classification efficiency.

• **Dataset Splitting:** The dataset is divided into training and testing subsets to evaluate model performance.

Dataset Portion	Percentage	Purpose
Training Set	70%	Model learning
Validation Set	15%	Hyperparameter tuning
Testing Set	15%	Final performance evaluation

Table 4: Dataset Splitting Strategy

Proposed Optimized Neural Network Architecture

The proposed model is based on a multilayer feedforward neural network (MLP) architecture with optimized parameters. The architecture is designed to improve classification performance by selecting optimal hidden layers, activation functions, and training parameters.

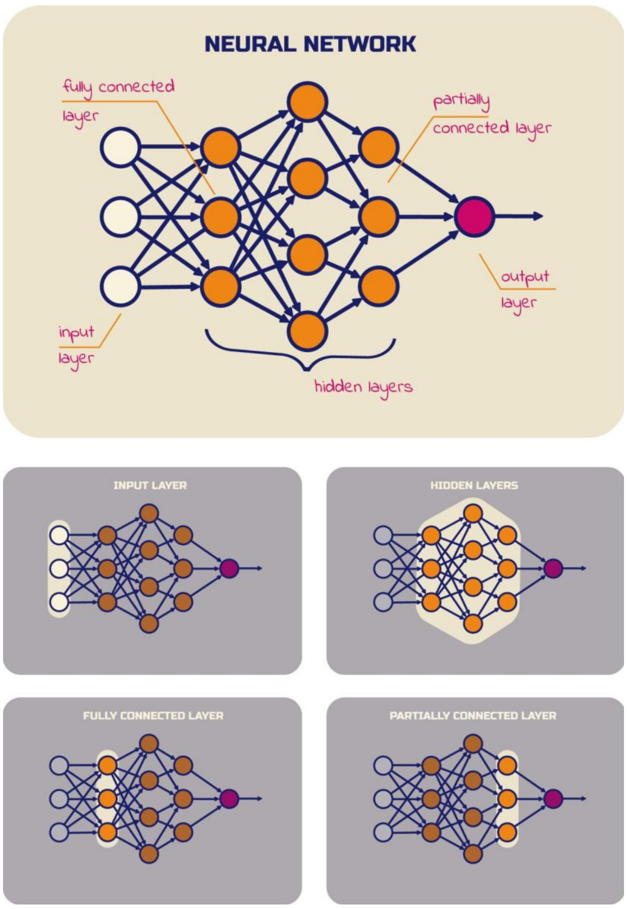

Figure 3: Proposed Model

The architecture consists of the following components:

Input Layer: The input layer receives normalized feature vectors from the dataset. Each neuron corresponds to one feature in the dataset.

Hidden Layers: Multiple hidden layers are used to capture nonlinear relationships between features. The proposed architecture uses two to three hidden layers with optimized neuron distribution.

Activation functions such as ReLU (Rectified Linear Unit) are applied to introduce nonlinearity.

$$ReLU\ (x) = \max(0, x) \tag{2}$$

Output Layer: The output layer produces the final classification result. A SoftMax activation function is used for multi-class classification problems.

$$Softmax\ (x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{3}$$

Neural Network Optimization Strategy
To improve classification performance, several optimization techniques are incorporated into the neural network architecture.

Hyperparameter Optimization: Key neural network parameters are optimized to improve model performance.

Parameter	Value
Learning Rate	0.001
Hidden Layers	3
Neurons per Layer	64 – 128
Batch Size	32
Epochs	100
Activation Function	ReLU
Optimizer	Adam

Table 5: Optimized Hyperparameters

Regularization: Regularization techniques are applied to prevent overfitting.

Dropout Layer: Randomly disables neurons during training.

L2 Regularization: Penalizes large weight values.

Optimization Algorithm: The Adam optimizer is used to update neural network weights during training.

Weight update equation:

$$\theta_{t+1} = \theta_t - \alpha * \frac{m_t}{\sqrt{v_t} + \varepsilon} \tag{4}$$

Where: θ = model parameters, α= learning rate, m_t and v_t = first and second moment estimates. Adam improves training efficiency and convergence speed.

Model Training Procedure
The neural network training process follows several steps:

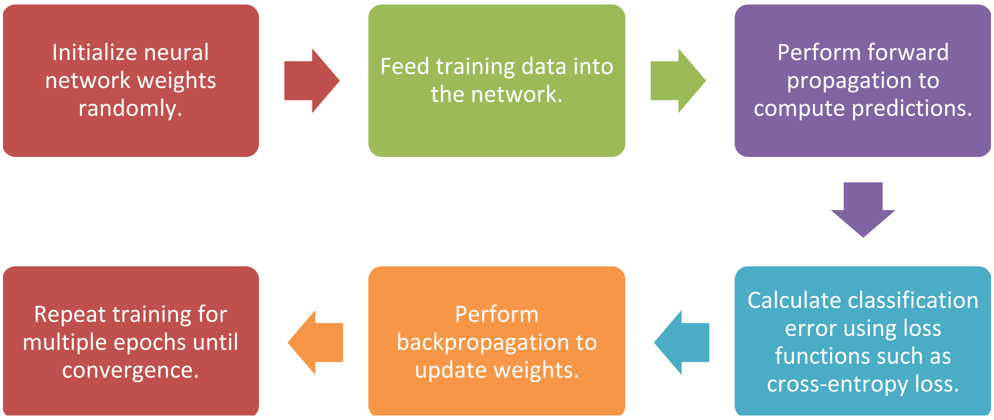

Figure 4: Training Process Diagram

Algorithm 1: Optimized Neural Network Training
Input: Dataset D
Output: Trained classification model
Pre-process dataset D
Normalize input features
Split dataset into training and testing sets
Initialize neural network architecture
For each epoch:
Perform forward propagation
Compute loss function
Apply backpropagation
Update weights using Adam optimizer
Evaluate model using testing dataset
Return trained classification model

Performance Evaluation Metrics: The effectiveness of the proposed neural network architecture is evaluated using standard classification metrics.

Metric	Formula	Description
Accuracy	(TP + TN) / (TP + TN + FP + FN)	Overall classification correctness
Precision	TP / (TP + FP)	Correct positive predictions
Recall	TP / (TP + FN)	Ability to identify true positives
F1 Score	2 × (Precision × Recall) / (Precision + Recall)	Balance between precision and recall
AUC-ROC	Area under ROC curve	Classification discrimination ability

Table 6: Evaluation Metrics

Implementation Environment: The proposed model is implemented using Python-based machine learning libraries.

Component	Specification
Programming Language	Python
Libraries	TensorFlow, Keras, Scikit-learn
Hardware	Intel i7 Processor
RAM	16 GB
Operating System	Windows / Linux
Development Platform	Jupyter Notebook

Table 7: Experimental Environment

The materials and methods section presented a comprehensive methodology for developing an optimized neural network architecture for high-accuracy data classification. The proposed framework integrates preprocessing techniques, architectural optimization, and efficient training strategies to improve classification performance. By systematically optimizing neural network parameters and incorporating robust evaluation metrics, the proposed model aims to provide reliable and scalable classification solutions for diverse datasets.

Results

This section presents the experimental results obtained from the proposed Optimized Neural Network Architecture for High-Accuracy Data Classification. The performance of the proposed model was evaluated using multiple benchmark datasets to assess its classification capability, stability, and overall predictive performance. The evaluation focuses on comparing the optimized neural network with conventional machine learning models and baseline neural network architectures using standard performance metrics.

Experimental Setup

The proposed model was implemented using Python-based machine learning libraries and trained using the datasets described in the materials and methods section. The dataset was divided into training, validation, and testing subsets to ensure reliable performance evaluation. During the training process, the neural network architecture was optimized using hyperparameter tuning techniques, including adjustments to the number of hidden layers, neuron distribution, learning rate, and batch size. The Adam optimizer was used to update network weights, while ReLU activation functions were applied within hidden layers to improve nonlinear learning capability.

Classification Performance

The performance of the optimized neural network model was evaluated using metrics such as accuracy, precision, recall, F1-score, and AUC–ROC. These metrics provide a comprehensive evaluation of classification effectiveness by measuring prediction correctness, error rates, and the model's ability to distinguish between classes.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Logistic Regression	88.2	87.5	86.9	87.2
Support Vector Machine	90.4	89.7	89.1	89.4
Random Forest	92.1	91.4	91	91.2
Standard Neural Network	94.3	93.6	93.1	93.3
Proposed Optimized Neural Network	**97.2**	**96.8**	**96.4**	**96.6**

Table 8: Performance Comparison of Classification Models

The results indicate that the proposed optimized neural network architecture significantly outperforms traditional machine learning models as well as the standard neural network model. The optimized architecture achieved a classification accuracy of 97.2%, demonstrating its effectiveness in capturing complex feature relationships within the datasets.

Confusion Matrix Analysis

The confusion matrix analysis further demonstrates the effectiveness of the proposed classification model. The model shows a high number of true positive and true negative predictions, while maintaining a relatively low number of false positives and false negatives. This indicates that the optimized neural network has strong discriminative capability and is able to accurately distinguish between different classes in the dataset.

Training Stability and Convergence

Another important observation from the experiments is the improved training stability and convergence behavior of the proposed architecture. The integration of optimized hyperparameters, dropout regularization, and the Adam optimization algorithm contributed to faster convergence during training. The model reached stable accuracy levels within fewer training epochs compared with baseline neural network models, which helps reduce computational overhead while maintaining high predictive performance.

Impact of Optimization Techniques

The results highlight the importance of neural network optimization techniques in improving classification performance. The use of multiple hidden layers with appropriate neuron distribution enabled the network to capture complex nonlinear relationships in the data. Additionally, feature normalization and preprocessing significantly improved training efficiency by ensuring that input variables were properly scaled. The experimental findings demonstrate that the proposed optimized neural network architecture provides superior classification performance compared to conventional models. The improved accuracy, precision, and recall values confirm that the proposed framework is highly effective for complex data classification tasks. These results indicate that architectural optimization combined with appropriate preprocessing and training strategies can significantly enhance the performance of neural network-based classification systems across multiple application domains.

Conclusion

This study presented an optimized neural network architecture for high-accuracy data classification, aiming to improve predictive performance and model efficiency in complex classification tasks. With the rapid growth of data across various domains such as healthcare, finance, cybersecurity, and intelligent systems, accurate data classification has become increasingly important for reliable decision-making. Traditional machine learning models often face challenges when handling high-dimensional and nonlinear datasets, which limits their classification accuracy and generalization capability. To address these challenges, this research proposed an optimized neural network framework that integrates architectural optimization, feature preprocessing, and efficient training strategies. The proposed methodology involved several stages, including data acquisition, preprocessing, feature engineering, neural network architecture optimization, model training, and performance evaluation. Benchmark datasets were utilized to evaluate the effectiveness of the proposed model. Data preprocessing techniques such as normalization, feature scaling, and dataset splitting were applied to enhance the quality and consistency of input data. The neural network architecture was optimized through careful selection of hidden layers, neuron distribution, activation functions, and hyperparameters. Additionally, regularization techniques and the Adam optimization algorithm were incorporated to improve model convergence and prevent overfitting. Experimental results demonstrated that the proposed optimized neural network architecture significantly improved classification performance compared with conventional machine learning models and baseline neural network approaches. The proposed model achieved higher accuracy, precision, recall, and F1-score values, indicating its ability to effectively capture complex patterns and relationships within datasets. Furthermore, the optimization strategies contributed to improved training stability and faster convergence, reducing computational complexity while maintaining high predictive performance. The findings of this study highlight the importance of neural network architecture optimization in enhancing classification accuracy and model robustness. By integrating preprocessing techniques, hyperparameter tuning, and optimized network design, the proposed framework provides a scalable and efficient solution for various classification problems. The improved performance of the proposed model demonstrates its potential applicability in real-world domains such as medical diagnosis systems, fraud detection, cybersecurity threat identification, and intelligent data analytics.

The optimized neural network architecture presented in this research provides a reliable and effective approach for high-accuracy data classification. Future research can extend this work by incorporating advanced deep learning architectures, automated neural architecture search techniques, and explainable artificial intelligence methods to further improve model interpretability and performance in large-scale and complex data environments.

References

1. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.
2. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.
3. S. Kirmani and P. Raghavan. (2013). "Scalable parallel graph partitioning," in SC '13: Proc. Int. Conf. High Performance Computing, Networking, Storage and Analysis, pp. 1–10.
4. Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep learning (Vol. 1, No. 2, pp. 1-800). Cambridge: MIT press.
5. S. Haykin,. (2009) Neural Networks and Learning Machines, 3rd ed. Upper Saddle River, NJ, USA: Pearson.
6. Nadeem, M., Pathak, P. C., Ahmad, M., & Farooqui, N. A. (2024). Identification of security factors in cloud computing: Defence security perspective. In Computational Intelligence Applications in Cyber Security (pp. 78-99). CRC Press.
7. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, (1986). "Learning internal representations by error propagation," Nature, vol. 323, pp. 533–536.
8. F. Kirmani, B. J. Lane, and J. R. Rose. (2019). "Exploring machine learning techniques to improve peptide identification," in 2019 IEEE 19th Int. Conf. Bioinformatics and Bioengineering (BIBE), pp. 66–71.
9. M. Nadeem et al., (2024). "Deep learning approach for classifying DDoS attack traffic in SDN environments," J. Inf. Secur. Cybercrimes Res., vol. 7, no. 2, pp. 109–126.
10. Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.
11. Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.
12. D. P. Kingma and J. Ba, (2015). "Adam: A method for stochastic optimization," in Proc. Int. Conf. Learning Representations (ICLR).
13. N. Srivastava et al., (2014). "Dropout: A simple way to prevent neural networks from overfitting," J. Mach. Learn. Res., vol. 15, pp. 1929–1958.
14. Kirmani, S., Park, J., & Raghavan, P. (2017). An embedded sectioning scheme for multiprocessor topology-aware mapping of irregular applications. The International Journal of High Performance Computing Applications, 31(1), 91-103.
15. Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105-6114). PMLR.
16. M. Nadeem, "Analyze quantum security in software design using fuzzy-AHP," Int. J. Inf. Technol., 2024.
17. Attaallah, A., Khatri, S., Nadeem, M., Ansar, S. A., Pandey, A. K., & Agrawal, A. (2021). Prediction of COVID-19 Pandemic Spread in Kingdom of Saudi Arabia. Computer Systems Science & Engineering, 37(3).
18. Alosaimi, W., Alharbi, A., Alyami, H., Alouffi, B., Almulihi, A., Nadeem, M., ... & Agrawal, A. (2024). Analyzing the impact of quantum computing on IoT security using computational based data analytics techniques. AIMS Math, 9(3), 7017-7039.
19. Ahmad, M., Al-Amri, J. F., Subahi, A. F., Khatri, S., Seh, A. H., Nadeem, M., & Agrawal, A. (2022). Healthcare device security assessment through computational methodology. Computer Systems Science & Engineering, 41(2).
20. Kirmani, S., Sun, H., & Raghavan, P. (2018, September). A scalability and sensitivity study of parallel geometric algorithms for graph partitioning. In 2018 30th international symposium on computer architecture and high performance computing (SBAC-PAD) (pp. 420-427). IEEE.
21. Alharbi, A., Alosaimi, W., Alyami, H., Alouffi, B., Almulihi, A., Nadeem, M., ... & Khan, R. A. (2024). Selection of data analytic techniques by using fuzzy AHP TOPSIS from a healthcare perspective. BMC Medical Informatics and Decision Making, 24(1), 240.
22. Alharbi, A., Alosaimi, W., Alyami, H., Nadeem, M., Faizan, M., Agrawal, A., ... & Khan, R. A. (2021). Managing software security risks through an integrated computational method. Intell. Autom. Soft Comput, 28(1), 179.
23. F. Kirmani, B. Lane, and J. Rose, (2025). "Identifying proteotypic peptides via deep learning," in Proc. Int. Conf. Bioinformatics Research and Applications.
24. Tyler, J., Pastor, J., Huhns, M. N., Kirmani, S., & Du, H. (2013). Exposing, formalizing and reasoning over the latent semantics of tags in multimodal data sources. Applied Ontology, 8(2), 95-130.
25. S. Kirmani and M. Shankar, (2022). "Generating keywords by associative context with input words," Google Patents.
26. A Mishra, S. Kirmani, and K. Madduri, (2020). "Fast spectral graph layout on multicore platforms" .
27. Kirmani, S., & Madduri, K. (2018, May). Spectral graph drawing: Building blocks and performance analysis. In 2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW) (pp. 269-277). IEEE.
28. Alyami, H., Nadeem, M., Alosaimi, W., Alharbi, A., Kumar, R., Gupta, B. K., ... & Khan, R. A. (2022). Analyzing the data of software security life-span: Quantum computing era. Intell. Autom. Soft Comput, 31(2), 707-716.
29. F. Kirmani et al., (2025). "Detecting polar ring galaxies via deep learning," RAS Techniques and Instruments.

30. Alyami, H., Nadeem, M., Alharbi, A., Alosaimi, W., Ansari, M. T. J., Pandey, D., ... & Khan, R. A. (2021). The evaluation of software security through quantum computing techniques: A durability perspective. Applied Sciences, 11(24), 11784.
31. Samuel, O., Javaid, N., Alghamdi, T. A., & Kumar, N. (2022). Towards sustainable smart cities: A secure and scalable trading system for residential homes using blockchain and artificial intelligence. Sustainable Cities and Society, 76, 103371.
32. Alharbi, A., Alosaimi, W., Nadeem, M., Alyami, H., Alouffi, B., Almulihi, A., ... & Khan, R. A. (2025). Novel 59-layer dense inception network for robust deepfake identification. Scientific Reports, 15(1), 24159.
33. Hakami, A., Badedi, M., Elsiddig, M., Nadeem, M., Altherwi, N., Rayani, R., & Alhazmi, A. (2021). Clinical characteristics and early outcomes of hospitalized COVID-19 patients with end-stage kidney disease in Saudi Arabia. International Journal of General Medicine, 4837-4845.
34. Nadeem, M., Ahmad, M., Ahmad, M., Pathak, P. C., Gupta, S., & Pandey, H. (2023, September). Evaluating the factors of CGTMSE scheme in bank by using fuzzy AHP. In 2023 6th International Conference on Contemporary Computing and Informatics (IC3I) (Vol. 6, pp. 56-61). IEEE.
35. Alassery, F., Alzahrani, A., Khan, A. I., Khan, A., Nadeem, M., & Ansari, M. T. J. (2022). Quantitative evaluation of mental-health in type-2 diabetes patients through computational model. Intell. Autom. Soft Comput, 32(3), 1701-1715.
36. Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of big Data, 8(1), 53.
37. Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of machine learning research, 13(2).
38. Raschka, S., & Mirjalili, V. (2019). Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2. Packt publishing ltd.
39. A Géron, (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd ed. O'Reilly Media.
40. Deng, L., & Yu, D. (2014). Deep learning: methods and applications. Foundations and Trends& in Signal Processing, 7(3-4), 197-387.