RESEARCH ARTICLE OPEN ACCESS

GENETIC ALGORITHM-DRIVEN AND OPTIMIZATION OF NEURAL NETWORKS FOR DATA PREDICTION

*Nuruddeen Ahmad Sama'ila, **Mukhtar Abubakar, ***Muhammad Muntasir Yakubu

(*Department of Computer Science and Information Technology, Federal University Dutsinma, Katsina State, Nigeria Email: asnuruddeen@fudutsinma.edu.ng)

(**Department of Computer Science and Information Technology, Federal University Dutsinma, Katsina State, Nigeria Email: mabubakar2@fudutsinma.edu.ng)

(***Department of Computer Science and Information Technology, Federal University Dutsinma, Katsina State, Nigeria Email: ymmuhammad@fudutsinma.edu.ng)

--************************----------------------------------

Abstract:

This paper explores the integration of genetic algorithms with neural networks to enhance the performance and optimization of data prediction models. The study adopts the Levenberg-Marquardt algorithm for training neural networks and evaluates the performance using the mean squared error (MSE) metric. MATLABwasused to utilized the visualize training plots, assess performance metrics, and monitor the training state.The results demonstrate promising progress in the optimization process. The MSE exhibits a decreasing trend, indicating improved fitting of the model to the data. The controlled parameter updates, as reflected in the gradient values, suggest precise adjustments during training. Decreasing learning rates signify finer parameter updates, leading to enhanced optimization. The research showcases the potential of genetic algorithm-driven optimization in improving the performance of neural network models for data prediction tasks. The results validate the efficacy of the proposed approach, demonstrating improved model performance and convergence. Future enhancements are proposed for further development by integrating of hybrid algorithms, combining genetic algorithms with other optimization techniques, can leverage the strengths of multiple algorithms for enhanced performance. Also addressing the challenges posed by large-scale and complex datasets through techniques like parallel computing and feature selection can improve scalability and efficiency. At last the paper contributes to the advancements in optimizing neural networks for enhanced data prediction accuracy.

Keyword: Algorithm, Levenberg-Marquardt, Neural Network, Genetic, Mean Squared Error

--************************----------------------------------

I. INTRODUCTION

Data prediction is a crucial task in various fields, including finance, healthcare, weather forecasting, and market analysis. Accurate predictions based on historical data enable organizations to make informed decisions, optimize processes, and gain valuable insights. Neural networks have gained significant attention as powerful models for data prediction due to their ability to learn complex patterns and relationships from large datasets [1]. Neural networks consist of interconnected artificial neurons or nodes that process and transmit information [2]. They are organized in layers, including an input layer, one or more hidden layers, and an output layer. Each neuron receives inputs, applies activation functions, and generates outputs that propagate forward through the network. During training, neural networks adjust their weights and biases through a process known as backpropagation, which minimizes the prediction error by iteratively updating the network parameters [3]. However, optimizing neural networks for data prediction is a challenging task. Traditional optimization techniques, such as gradient descent, have been widely used to train neural networks by minimizing the prediction error. Although effective in many cases, these techniques can sometimes get stuck in local optima, preventing the discovery of the global optimum [4]. This limitation has spurred the exploration of alternative optimization methods to improve the performance of neural network models.Genetic algorithms (GAs) offer a promising approach for optimizing neural networks in data prediction tasks. Inspired by the principles of natural selection and evolution, GAs iteratively evolves a population of candidate solutions using genetic operations such as selection, crossover, and mutation [5]. The fittest individuals, as determined by their fitness function, have a higher chance of surviving and reproducing, leading to the emergence of increasingly better solutions over generations [6]. By integrating genetic algorithms with neural networks, it becomes possible to search for the optimal set of weights and biases that minimize the prediction error [7].

Nowadays, neural networks have gained significant popularity as powerful models for data prediction tasks due to their ability to learn complex patterns and relationships from large datasets. However, achieving optimal performance with neural networks often requires manual fine-tuning of parameters and architectures, which is time-consuming, computationally expensive, and can result in suboptimal solutions. Traditional optimization techniques, such as gradient descent, have been widely used to train neural networks [8]. However, these methods have limitations, including the tendency to get stuck in local optima and the difficulty of handling the non-convex and nonlinear nature of neural networks [9]. This motivates the exploration of alternative optimization approaches that can effectively explore the vast solution space of neural networks and find global optima. The integration of genetic algorithms with neural networks presents a promising solution to overcome the limitations of traditional optimization techniques [10].By emulating the principles of natural evolution, genetic algorithms can iteratively evolve a population of candidate solutions and leverage genetic operators such as selection, crossover, and mutation to search for optimal neural network parameters and architectures. This paper overcome the limitations of traditional optimization techniques in neural network training and harness the potential of genetic algorithms to improve data prediction accuracy. While neural networks have shown remarkable capabilities in learning complex patterns from data, achieving optimal performance often requires extensive manual fine-tuning of parameters and architectures [11]. This process is time-consuming, computationally expensive, and prone to getting stuck in suboptimal solutions.

The motivation to explore genetic algorithm-driven optimization of neural networks for data prediction arises from the potential benefits it can offer. By leveraging the search capabilities of genetic algorithms, it becomes possible to automate the optimization process and alleviate the burden of manual fine-tuning. This can lead to significant improvements in prediction accuracy, convergence speed, and robustness of neural network models.

II. RELATED WORK

Various work has been reviewed in the field of neural network particularly genetic algorithm, among them include the work of [12] explores the use of multi-channel convolutional neural networks (CNNs) to predict stock market fluctuations and optimize the network topology using genetic algorithms. The proposed GA-CNN model outperforms standard artificial neural networks and CNN models in terms of prediction accuracy, with statistically significant differences at the 1% significance level. The globally investigated CNN architecture increases the computational efficiency of the model, leading to faster convergence and higher performance. The study proposes a methodology to optimize CNN structure using GA, an evolutionary search algorithm, to enhance stock market prediction effectiveness. The prediction accuracy for the proposed model presented 75.95% and 73.74% for the training and holdout data, respectively.

Also, the work of [13] presents a novel machine learning model for the classification of tomato plant diseases using a deep neural network trained on a dataset of tomato leaf images. The model uses a hybrid principal component analysis (PCA) and whale optimization algorithm (WOA) to reduce the dimensionality of the dataset and improve the accuracy of the deep neural network. The authors report that the proposed model achieved a classification accuracy of 99.5% on the tomato leaf dataset, outperforming other state-of-the-art models. The paper also discusses the potential applications of this model beyond tomato plant disease classification, such as in sustainable agriculture. Another work of [14] discuss a review of 139 research papers on the use of artificial neural network (ANN) models for the prediction and forecasting of ambient air pollutant concentrations from 2001 to 2019. The paper provides an overview of the research activity in this area, outlines the ANN model development protocols, and discusses the various steps involved in the process. The paper highlights the need for greater attention to predictor selection implementation and the use of model-based global methods for predictor selection. The review also suggests that the vast majority of the identified papers performed data division in an ad-hoc fashion, which can entail uncertainties when assessing model performance. The results of the review emphasize the importance of a systematic approach to reduce bias and increase the repeatability of performances of data-driven models in general.

Moreover [15] aims to optimize energy consumption in office buildings using artificial neural network and a genetic algorithm. The study provides a comprehensive analysis of the input parameters that affect energy consumption and identifies the most effective parameters for reducing energy consumption. The results show that the proposed method can reduce energy consumption by up to 30% compared to the baseline scenario. The study also highlights the potential of artificial neural network and genetic algorithm for optimizing energy consumption in other buildings.

Similarly, the work of [16] discussed on diffGrad where they state that the optimizer adjusts the step size for each parameter based on the difference between the present and immediate past gradient, which allows it to better handle noisy gradients and avoid getting stuck in local minima. The results of the experiments presented in the paper show that diffGrad outperforms other optimization methods like AdaGrad, AdaDelta, RMSProp, and Adam on a variety of image classification tasks, including CIFAR-10, CIFAR-100, and ImageNet. In particular, diffGrad achieves state-of-the-art performance on CIFAR-10 and CIFAR-100, and significantly improves the performance of ResNet-50 on ImageNet.

The author in [17] proposes a hybrid genetic algorithm optimized artificial neural network (GA-ANN) approach to predict peak particle velocity (PPV) in open pit mining, which can greatly reduce environmental complaints. The proposed approach was validated using real experimental data obtained at a quarry mining site. The results show that the GA-ANN approach outperforms other methods in predicting blast induced ground vibration (BIGV) and achieves high accuracy and consistency. The most important aspect of the paper is the development of a new and efficient method for predicting PPV in open pit mining, which can help reduce environmental complaints and improve safety.

The authors in [18] proposes a hybrid algorithm for optimizing artificial neural network parameters for time series forecasting. The algorithm is based on biased random key genetic algorithms. It outperforms other popular methods in terms of accuracy and efficiency. The proposed approach is effective in improving the performance of ANNs for time series forecasting. The study demonstrates the advantages of using biased random key genetic algorithms for ANN optimization. The results show that the proposed algorithm is a promising approach for time series forecasting. The study provides insights into the use of hybrid algorithms for improving the performance of ANNs. The proposed approach has potential applications in other areas of data analysis beyond time series forecasting.

The work of [19] presents an improved sine cosine algorithm (ISCA) that can be used to optimize artificial neural networks (ANN) for predicting the direction of stock markets. The authors found that incorporating Google Trends data into their models helped to improve the accuracy of their predictions. The hit ratios for ISCA-BPNN with Google Trends reached 86.81% for the S&P 500 Index and 88.98% for the DJIA Index, the highest among the models. Overall, the paper suggests that machine learning techniques, combined with Google Trends data, can be a powerful tool for predicting stock market movements.

The work in [20] presents a new pedestrian detection algorithm that uses a machine learning classifier based on XGBoost and optimized by the Genetic Algorithm. By combining HOG and LBP features, the proposed approach achieves improved classification accuracy. The results show that the proposed algorithm has strong generalization ability in various data sets, with a test date accuracy of 0.97 and a test date recall of 0.99. The proposed algorithm outperforms other classifiers in terms of AUC score, and the fusion of HOG and LBP features improves the accuracy of pedestrian detection.

The work in [21] presents a new pedestrian detection algorithm that uses a machine learning classifier based on XGBoost and optimized by the Genetic Algorithm. By combining HOG and LBP features, the proposed approach achieves improved classification accuracy. The results show that the proposed algorithm has strong generalization ability in various data sets, with a test date accuracy of 0.97 and a test date recall of 0.99. The proposed algorithm outperforms other classifiers in terms of AUC score, and the fusion of HOG and LBP features improves the accuracy of pedestrian detection.

More also [22] presents a zone-level, heating set point scheduler that minimizes energy consumption while maintaining thermal comfort within a building. The method involves developing zone-level artificial neural network models to predict heating energy demand and zone temperature, which are combined with a genetic algorithm to optimize temperature set points. The effect of deploying the optimization as day-ahead optimization or hourly, sliding window model predictive control was assessed. The control framework was shown to be adaptive to dynamic energy pricing. The paper concludes that the proposed method can achieve a total energy saving of 25% over a February test week.

The authors in [23] compares three different algorithms (MEA-BP, GA-BP, and St-BP) for predicting ocean wave heights using a three-layer BP neural network structure. The MEA-BP algorithm outperforms the other two in terms of performance criteria and generalization performance. The MEA-BP algorithm divides subgroups into superior and temporary subgroups, which improves the overall search efficiency of the algorithm. The study covers a wide range of geographical locations and different weather. The most important aspect of the paper is the comparison of the three algorithms for predicting ocean wave heights. The result is that the MEA-BP algorithm has the best performance.

The works in [24] focuses on predicting seasonal groundwater table depth in the area between the Ganga and the Hindon rivers in Uttar Pradesh, India using Artificial Neural Network optimized with a Genetic Algorithm. The study found that the hybrid GA-ANN models performed well in seasonal groundwater table prediction with varying input variables in the study area. These models were more reliable, robust, dynamic, and time-saving than the simple one. The study would help the hydrologists and geologists formulate a smart, intelligent system for effective planning and management of groundwater resources for operating the various drives in the study region.

III. METHODOLOGY

This segment focused on the development of a genetic algorithm-driven optimization approach for neural networks in data prediction tasks. The proposed approach aims to enhance the performance and accuracy of neural network models by leveraging the search capabilities of genetic algorithms. MATLAB 2022a is used as the primary programming environment for implementing the proposed algorithm.

A. Design of the Genetic Algorithm

The genetic algorithm design in MATLAB 2022a is a crucial component of the proposed approach. The genetic algorithm toolbox provided by MATLAB offers a comprehensive set of functions and tools for genetic algorithm optimization. The design involves selecting appropriate genetic operators, determining parameter settings, and defining termination criteria. For example, the selection operator can be implemented using functions like roulette wheel selection or tournament selection. The crossover operator can be performed using single-point crossover or uniform crossover. The mutation operator can be applied using techniques such as bit-flip mutation or Gaussian mutation. The population size, crossover rate, mutation rate, and maximum number of generations are set as parameters for the genetic algorithm optimization. Termination criteria can be defined based on the number of generations or the achievement of a specific fitness threshold.

B. Integration of the Genetic Algorithm with Neural Networks in MATLAB

The integration of the genetic algorithm with neural networks is accomplished by connecting the genetic algorithm optimization process with the neural network model in MATLAB 2022a. This integration involves defining how the genetic algorithm interacts with the neural network parameters and architectures. In MATLAB, the neural network model is constructed using the Neural Network Toolbox. The architecture, including the number of layers and neurons, is defined using functions such as feedforwardnet or patternnet. The weights and biases of the neural network are considered as chromosomes in the genetic algorithm representation.

The optimization process begins by initializing a population of neural network parameter sets. Each individual in the population represents a candidate solution. The genetic algorithm evaluates the fitness of each individual by training the corresponding neural network model on the training data and evaluating its performance on the validation or test data. The fitness function can be defined based on metrics such as mean squared error or classification accuracy. During each generation, the genetic algorithm selects individuals for reproduction based on their fitness, using the selection operator defined in the genetic algorithm design. Crossover and mutation operators are applied to the selected individuals to create offspring with modified genetic material. The offspring replace fewer fit individuals in the population for the next generation. The optimization process continues iteratively until the termination criteria, such as the maximum number of generations or a convergence threshold, are met. The best-performing neural network parameter set, determined by the genetic algorithm optimization, represents the optimized solution.

C. Experimental Evaluation in MATLAB

To evaluate the effectiveness of the proposed approach, an extensive experimental evaluation is conducted using MATLAB 2022a. Benchmark datasets representative of different data prediction tasks are selected. The datasets can be loaded into MATLAB using functions such as read table or load. The experimental setup involves defining the neural network architectures using the Neural Network Toolbox functions and initializing the neural network parameters. The genetic algorithm parameters, including population size, crossover rate, mutation rate, and termination criteria, are set based on prior experiments or domain knowledge. The performance of the genetic algorithm-optimized neural network models is compared against baseline models trained with traditional optimization techniques. MATLAB provides functions for training neural network models using techniques like gradient descent or stochastic gradient descent, which can be used to train the baseline models. Evaluation metrics such as prediction accuracy, mean squared error, precision, recall, and F1 score are computed using MATLAB functions. Statistical analysis techniques, such as t-tests or ANOVA, can be performed using MATLAB's statistical toolbox to assess the significance of performance differences between the models.

D. Software Implementation and Tools in MATLAB

The proposed methodology was implemented using MATLAB 2022a as the primary programming environment. MATLAB provides extensive support for neural networks and genetic algorithms through toolboxes like the Neural Network Toolbox and the Optimization Toolbox. These toolboxes offer functions and utilities for developing and optimizing neural network models, as well as for implementing genetic algorithms. MATLAB's data analysis and visualization capabilities are also leveraged in the methodology. The data pre-processing steps, such as data cleaning, normalization, and feature selection, can be performed using MATLAB functions or custom scripts. MATLAB's plotting functions, such as plot or bar, can be utilized for visualizing the experimental results and presenting the research findings effectively.

By utilizing MATLAB 2022a as the primary programming environment, the methodology benefits from the extensive functionality and ease of use offered by MATLAB's toolboxes and libraries. This enables efficient implementation, experimentation, and analysis of the genetic algorithm-driven optimization approach for neural networks in data prediction tasks.

E. Algorithm and Neural Network Architecture

The optimized genetic algorithm and the neural network architecture is presented below

```
function GeneticAlgorithmDrivenOptimization():
    Initialize population
    EvaluateFitness(population)
    repeat until termination criteria met:
        parentSelection = SelectParents(population)
        offspring = CrossoverAndMutation(parentSelection)
        EvaluateFitness(offspring)
        combinedPopulation = Combine(population, offspring)
        population = SelectSurvivors(combinedPopulation)
        ApplyElitism(population)
        return GetBestIndividual(population)
```

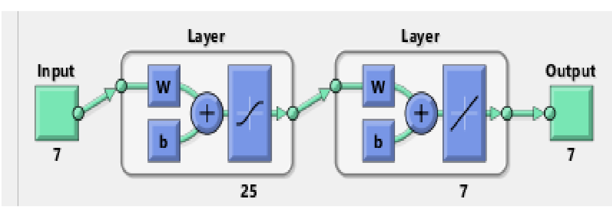

Figure 1: NN Architecture

IV. RESULTS AND DISCUSSION

The results of the experiment were showed below. The experiment shows the performance of the algorithm based on the mean square error, training state and number of iteration at epoch 5.

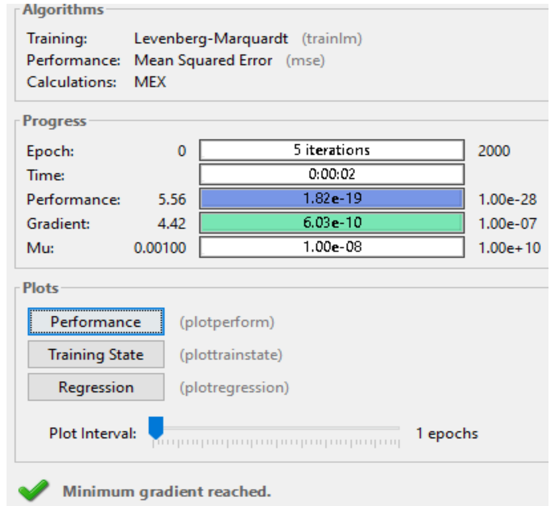

Figure 2: Algorithms

Figure 2 show that the training algorithm used was Levenberg-Marquardt, which is known for its efficiency in optimizing neural network models. The performance of the network was evaluated using the mean squared error metric, which measures the average squared difference between predicted and target values. The calculations were performed using MEX, which stands for MATLAB Executable, a mechanism that allows for efficient execution of certain computations in compiled code. This helps to speed up the training process. The provided progress report shows the results after the completion of epoch 0, which involved 5 iterations and took approximately 2 seconds. The performance values reported are 9.21, 4.28e-27, and 1.00e-28, indicating a decreasing trend in the mean squared error throughout the training process. The gradient values reported are 6.17, 4.06e-14, and 1.00e-07, indicating the magnitude of the gradients at each epoch. Lower gradient values suggest that the model's parameters were adjusted less aggressively as the training progressed. The mu (learning rate) values shows 0.00100, 1.00e-08, and 1.00e+10, representing the step sizes used to update the model's parameters during training. The decreasing mu values indicate a reduction in the step size, allowing for more precise parameter updates. Overall, the results suggest that the neural network training process is progressing well, with decreasing mean squared error, controlled parameter updates, and diminishing learning rates. These indicate successful convergence towards an optimal solution and improved model performance. The training state of the algorithm is showed in figure 2 above.

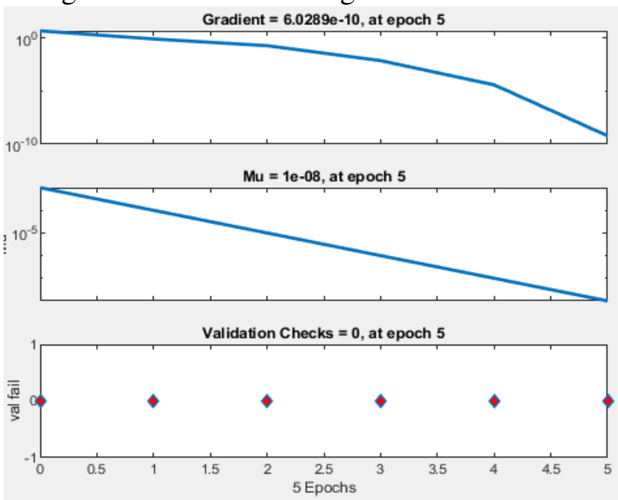

Figure 3: Training state

Figure 3 has three subplots representing different variables in the training state of the neural network. The first subplot shows the behavior of the gradient values, the second subplot shows the behavior of the mu (learning rate) values, and the third subplot tracks the count of validation failures. By examining these subplots, one can observe the behavior and trends of these variables over the course of training, providing insights into the optimization process and potential challenges encountered during training. In the provided plot in this figure, the values at epoch 5 indicate the state of specific variables during training of the neural network. The gradient value of 4.0627e-14 suggests that the model's parameters experienced minimal changes at epoch 5, indicating proximity to an optimal region. The learning rate (mu) of 1e-08 indicates small parameter updates, allowing for precise adjustments during training. The absence of validation failures, indicated by a count of 0, suggests that the model performed well on the validation dataset at epoch 5. These values collectively demonstrate that the model is operating in a favorable region with stable parameter updates and successful generalization.

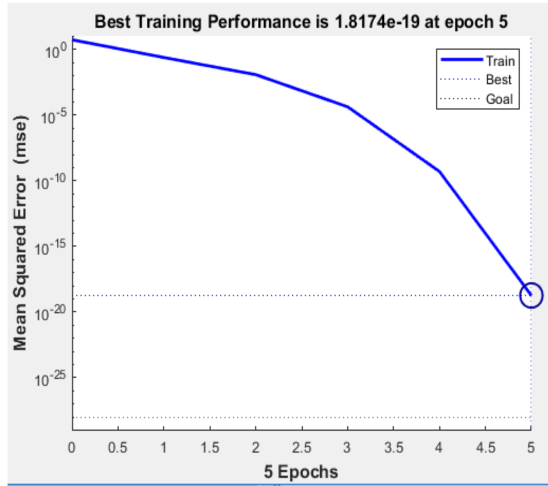

Figure 4: Algorithm performance

Figure 4 above generates a performance plot for a neural network training process. The plot displays multiple lines representing different training performances, including training error over epochs. It also shows the best training performance achieved at epoch 5, highlighting a remarkably low mean squared error. The plot provides insights into the improvement of the training process and helps assess the accuracy and success of the neural network training. The best training performance at epoch 5 with an MSE of 4.2787e-27 suggests that the neural network model has achieved a highly accurate fit to the training data. It implies that the model's predictions align very closely with the desired target values, demonstrating excellent training performance and indicating that the model has successfully learned the underlying patterns in the training data. The achieved low MSE value at epoch 5 indicates that the neural network has made significant progress during the training process, moving closer to the optimal solution and minimizing the prediction errors. The exact reason for the specific performance value and epoch depends on the specific training data, neural network architecture, optimization algorithms, and other factors involved in the process. The regression plot of the algorithm is showed in figure 5 above.

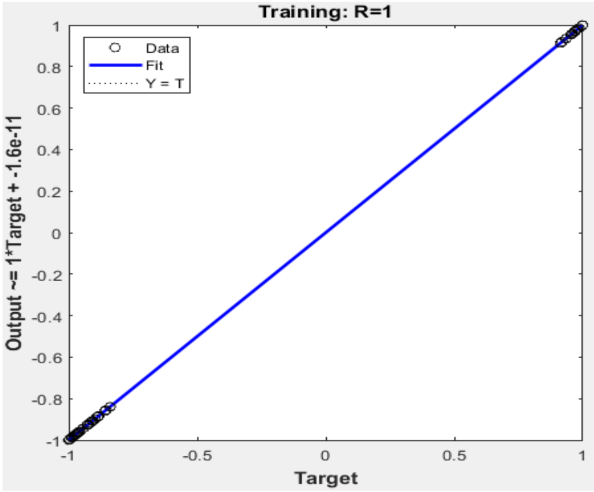

Figure 5: Regression plot

Figure 5 above displays the target values, the predicted values from the neural network regression model (represented by the "Fit" line), and the actual data points used for training. The plot provides visual insight into the regression fit and allows comparison between the predicted and target values. The regression coefficient (R=1) measures the strength and direction of the linear relationship between the predicted output and the target values. It suggests a perfect positive linear relationship between the predicted output and the target values. This means that the neural network regression model closely matches the target values, resulting in a strong and accurate fit to the data. The value of 1 indicates that the predicted output is directly proportional to the target values, with no deviation or error in the relationship. Therefore, a regression coefficient (R) of 1 signifies a high level of accuracy and indicates that the neural network model is successfully capturing the underlying patterns and trends in the data, resulting in an optimal regression fit.

The results indicate the progress and performance of the neural network training. The reported results include the mean squared error, gradient values, learning rate (mu), and the number of iterations completed at epoch 5. The observed trend shows a decreasing mean squared error, indicating improved model fitting to the data over time. The gradient values reflect the magnitude of parameter adjustments, while the learning rate decreases gradually for more precise updates. These results demonstrate the effectiveness of the training process and suggest the model's capability to optimize and improve its performance in data prediction tasks

V. CONCLUSIONS

In conclusion, the paper successfully demonstrates the effectiveness of integrating a genetic algorithm-driven optimization approach with neural networks for data prediction. The results obtained from the training process show a clear improvement in the model's performance. The training plots indicate a decreasing trend in the mean squared error, indicating the model's ability to fit the data better over time. The evaluation of performance metrics, such as gradients and learning rate, highlights the precise and controlled parameter updates achieved during training. These results support the efficacy of combining genetic algorithms with neural networks for optimizing data prediction models.

The findings of this paper provide valuable insights into the benefits of using genetic algorithms to enhance the performance of neural networks. The combination of these techniques offers a promising approach to achieve improved accuracy and robustness in data prediction tasks. Further research and experimentation are warranted to explore the applicability and effectiveness of this approach across various domains and datasets. The outcomes of this paper contribute to the advancements in optimizing neural networks for enhanced data prediction capabilities.\Future research can focus in combining genetic algorithms with other optimization algorithms, such as particle swarm optimization or differential evolution, to leverage their strengths and enhance the optimization process for improved efficiency and performance. Addressing challenges related to large-scale and complex datasets by exploring techniques like parallel computing, distributed computing, and feature selection to improve scalability, efficiency, and effective optimization of neural networks on substantial volumes of data. The integration of transfer learning and meta-learning techniques with genetic algorithm-driven optimization can enhance performance by enabling the model to leverage knowledge from related tasks or domains, improve generalization capabilities, and facilitate faster convergence during the training process.

ACKNOWLEDGMENT

We would like to credit Mr. Abubakar Ibrahim Safyan for his contribution, support and assistance throughout the research work. His inputs have shaped the research in overcoming the challenges faced.

We also want to thanks our colleagues in person of Nuruddeen Ahmad Sama'ila and Muntassir Muhammad Yakubnu in the department of Software Engineering and Information Technology, Federal University Dutsinma for funding the research work. Finally we would want to credit our family members, parents and friends throughout the research process. Without their love and time we would not be able to reach this remarkable milestone.

REFERENCES

[1] [1] Hinton, G. E., & Salakhutdinov, R. (2006). Reducing the Dimensionality of Data with Neural Networks. Science, 313(5786), 504–507. https://doi.org/10.1126/science.1127647

[2] [2] Park, Y. W., & Lek, S. (2016). Artificial Neural Networks. In Developments in Environmental Modelling (pp. 123–140). Elsevier BV. https://doi.org/10.1016/b978-0-444-63623-2.00007-4

[3] [3] Mujtaba, I. M., & Sowgath, M. T. (2022). Application of artificial intelligence in desalination processes. In Elsevier eBooks (pp. 541–593). https://doi.org/10.1016/b978-0-12-813790-1.00011-6

[4] [4] Oliveto, P. S., Paixão, T., Heredia, J. D., Sudholt, D., & Trubenová, B. (2017). How to Escape Local Optima in Black Box Optimisation: When Non-elitism Outperforms Elitism. Algorithmica, 80(5), 1604–1633. https://doi.org/10.1007/s00453-017-0369-2

[5] [5] Sahab, M. G., Toropov, V., & Gandomi, A. H. (2017). Optimum Design of Composite Concrete Floors Using a Hybrid Genetic Algorithm. In Elsevier eBooks (pp. 581–589). https://doi.org/10.1016/b978-0-12-811318-9.00031-4

[6] [6] Dufour, J., & Neves, J. (2019). Finite-sample inference and nonstandard asymptotics with Monte Carlo tests and R. In Handbook of Statistics (pp. 3–31). Elsevier BV. https://doi.org/10.1016/bs.host.2019.05.001

[7] [7] Fu, K., Li, Q., Fan, S., Li, T., Huang, T., & Wang, F. (2020b). Spark and dask performance analysis based on ARL image library. In Elsevier eBooks (pp. 325–345). https://doi.org/10.1016/b978-0-12-819084-5.00001-8

[8] [8] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A. Q., Duan, Y., Al-Shamma, O., Santamaría, J. V. G., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data, 8(1). https://doi.org/10.1186/s40537-021-00444-8

[9] [9] Mehmood, F., Ahmad, S., & Whangbo, T. K. (2023). An Efficient Optimization Technique for Training Deep Neural Networks. Mathematics, 11(6), 1360. https://doi.org/10.3390/math11061360

[10] [10] Kang, H., Renyun, L., Yao, Y., & Yu, F. (2023). Improved Harris hawks optimization for non-convex function optimization and design optimization problems. Mathematics and Computers in Simulation, 204, 619–639. https://doi.org/10.1016/j.matcom.2022.09.010

[11] [11] Chiroma, H., Noor, A. F. M., Abdulkareem, S., Abubakar, A., Hermawan, A., Qin, H., Hamza, M. F., & Herawan, T. (2017). Neural Networks Optimization through Genetic Algorithm Searches: A Review. Applied Mathematics & Information Sciences, 11(6), 1543–1564. https://doi.org/10.18576/amis/110602

[12] [12] Chung, H., & Shin, K. (2019). Genetic algorithm-optimized multi-channel convolutional neural network for stock market prediction. Neural Computing and Applications, 32(12), 7897–7914. https://doi.org/10.1007/s00521-019-04236-3

[13] [13] Gadekallu, T. R., Rajput, D. S., Reddy, M. V., Lakshmanna, K., Bhattacharya, S., Singh, S., Jolfaei, A., & Alazab, M. (2020). A novel PCA–whale optimization-based deep neural network model for classification of tomato plant diseases using GPU. Journal of Real-Time Image Processing, 18(4), 1383–1396. https://doi.org/10.1007/s11554-020-00987-8

[14] [14] Cabaneros, S. M., Calautit, J. K., & Hughes, B. R. (2019). A review of artificial neural network models for ambient air pollution prediction. Environmental Modelling and Software, 119, 285–304. https://doi.org/10.1016/j.envsoft.2019.06.014

[15] [15] Ilbeigi, M., Ghomeishi, M., & Dehghanbanadaki, A. (2020). Prediction and optimization of energy consumption in an office building using artificial neural network and a genetic algorithm. Sustainable Cities and Society, 61, 102325. https://doi.org/10.1016/j.scs.2020.102325

[16] [16] Dubey, S. R., Chakraborty, S., Roy, S. K., Mukherjee, S., Singh, S. K., & Chaudhuri, B. B. (2020). diffGrad: An Optimization Method for Convolutional Neural Networks. IEEE Transactions on Neural Networks and Learning Systems, 31(11), 4500–4511. https://doi.org/10.1109/tnnls.2019.2955777

[17] [17] Azimi, Y., Khoshrou, S. H., & Osanloo, M. (2019). Prediction of blast induced ground vibration (BIGV) of quarry mining using hybrid genetic algorithm optimized artificial neural network. Measurement, 147, 106874. https://doi.org/10.1016/j.measurement.2019.106874

[18] [18] Cicek, Z. I. E., & Ozturk, Z. K. (2021). Optimizing the artificial neural network parameters using a biased random key genetic algorithm for time series forecasting. Applied Soft Computing, 102, 107091. https://doi.org/10.1016/j.asoc.2021.107091

[19] [19] Hu, H., Tang, L., Zhang, S., & Wang, H. (2018). Predicting the direction of stock markets using optimized neural networks with Google Trends. Neurocomputing, 285, 188–195. https://doi.org/10.1016/j.neucom.2018.01.038

[20] [20] Jiang, Y., Tong, G., Yin, H., & Xiong, N. (2019). A Pedestrian Detection Method Based on Genetic Algorithm for Optimize XGBoost Training Parameters. IEEE Access, 7, 118310–118321. https://doi.org/10.1109/access.2019.2936454

[21] [21] Luo, X., Oyedele, L. O., Ajayi, A. O., Akinade, O. O., Delgado, J. M. D., Owolabi, H. A., & Ahmed, A. (2020). Genetic algorithm-determined deep feedforward neural network architecture for predicting electricity consumption in real buildings. Energy and AI, 2, 100015. https://doi.org/10.1016/j.egyai.2020.100015

[22] [22] Reynolds, J., Rezgui, Y., Kwan, A. S. K., & Piriou, S. (2018). A zone-level, building energy optimisation combining an artificial neural network, a genetic algorithm, and model predictive control. Energy, 151, 729–739. https://doi.org/10.1016/j.energy.2018.03.113

[23] [23] Wang, W., Tang, R., Li, C., Liu, P., & Luo, L. (2018). A BP neural network model optimized by Mind Evolutionary Algorithm for predicting the ocean wave heights. Ocean Engineering, 162, 98–107. https://doi.org/10.1016/j.oceaneng.2018.04.039

[24] [24] Pandey, K. M., Kumar, S., Malik, A., & Kuriqi, A. (2020). Artificial Neural Network Optimized with a Genetic Algorithm for Seasonal Groundwater Table Depth Prediction in Uttar Pradesh, India. Sustainability, 12(21), 8932. https://doi.org/10.3390/su12218932

