Deep convolutional neural network and IoT technology for healthcare

Sobia Wassan[1] (iD) **, Hu Dongyan**[2] **, Beenish Suhail**[3] **, N.Z. Jhanjhi**[4] (iD) **,
Guanghua Xiao**[5] **, Suhail Ahmed**[6] **and Raja Kumar Murugesan**[4]

Abstract

Background: Deep Learning is an AI technology that trains computers to analyze data in an approach similar to the human brain. Deep learning algorithms can find complex patterns in images, text, audio, and other data types to provide accurate predictions and conclusions. Neuronal networks are another name for Deep Learning. These layers are the input, the hidden, and the output of a deep learning model. First, data is taken in by the input layer, and then it is processed by the output layer. Deep Learning has many advantages over traditional machine learning algorithms like a KA-nearest neighbor, support vector algorithms, and regression approaches. Deep learning models can read more complex data than traditional machine learning methods.

Objectives: This research aims to find the ideal number of best-hidden layers for the neural network and different activation function variations. The article also thoroughly analyzes how various frameworks can be used to create a comparison or fast neural networks. The final goal of the article is to investigate all such innovative techniques that allow us to speed up the training of neural networks without losing accuracy.

Methods: A sample data Set from 2001 was collected by www.Kaggle.com. We can reduce the total number of layers in the deep learning model. This will enable us to use our time. To perform the ReLU activation, we will make use of two layers that are completely connected. If the value being supplied is larger than zero, the ReLU activation will return 0, and else it will output the value being input directly.

Results: We use multiple parameters to determine the most effective method to test how well our method works. In the next paragraph, we'll discuss how the calculation changes secret-shared Values. By adopting 19 train set features, we train our reliable model to predict healthcare cost's (numerical) target feature. We found that 0.89503 was the best choice because it gave us a good fit (R2) and let us set enough coefficients to 0. To develop our stable model with this Set of parameters, we require 26 iterations. We use an R2 of 0.89503, an MSE of 0.01094, an RMSE of 0.10458, a mean residual deviance of 0.01094, a mean absolute error of 0.07452, and a root mean squared log error of 0.07207. After training the model on the train set, we applied the same parameters to the test set and obtained an R2 of 0.90707, MSE of 0.01045, RMSE of 0.10224, mean residual deviation of 0.01045, MAE of 0.06954, and RMSE of 0.07051, validating our solution approach. The objective value of our secured model is higher than that of the scikit-learn model, although the former performs better on goodness-of-fit criteria. As a result, our protected model performs quite well, marginally outperforming the (very optimized) scikit-learn model. Using a backpropagation algorithm and stochastic gradient descent, deep Learning develops artificial neural systems with several interconnected layers. There may be hidden layers of neurons in the network that have the tanh, rectification, and max-out hyperparameters. Modern features like momentum training, dropout, active learning rate, rate annealed, and L1 or L2 regularization provide exceptional prediction performance. The worldwide model's parameters are multi-threadedly (asynchronously) trained on the data from that node, and the model-based data is then gradually augmented by model averaging over the entire network. The method is executed on a single-node, direct H2O cluster initiated by the operator. The

[1]School of Equipment Engineering, Jiangsu Urban and Rural Construction Vocational College, Changzhou, China
[2]Department of Research and Industry, Jiangsu Urban and Rural Construction Vocational College, Changzhou, China
[3]School of Economics, Shanghai University, Shanghai, China
[4]School of Computer Science, SCS, Taylor's University, Subang Jaya, Malaysia

[5]School of Equipment Engineering, Jiangsu Urban and Rural Construction College, Changzhou, Jiangsu, China
[6]School of Economics, University of Sindh, Jamshoro, Sindh, Pakistan

Corresponding author:
N.Z. Jhanjhi, School of Computer Science, SCS, Taylor's University, Subang Jaya 47500, Malaysia.
Email: noorzaman.jhanjhi@taylors.edu.my

operation is parallel despite there just being a single node involved. The number of threads may be adjusted in the settings menu under Preferences and General. The optimal number of threads for the system is used automatically. Successful predictions in the healthcare data sets are made using the H2O Deep Learning operator. There will be a classification done since its label is binomial. The Splitting Validation operator creates test and training datasets to evaluate the model. By default, the settings of the Deep Learning activator are used. To put it another way, we'll construct two hidden layers, each containing 50 neurons. The Accuracy measure is computed by linking the annotated Sample Set with a Performer (Binominal Classification) operator. Table 3 displays the Deep Learning Model, the labeled data, and the Performance Vector that resulted from the technique.

Conclusions: Deep learning algorithms can be used to design systems that report data on patients and deliver warnings to medical applications or electronic health information if there are changes in the patient's health. These systems could be created using deep Learning. This helps verify that patients get the proper effective care at the proper time for each specific patient. A healthcare decision support system was presented using the Internet of Things and deep learning methods. In the proposed system, we examined the capability of integrating deep learning technology into automatic diagnosis and IoT capabilities for faster message exchange over the Internet. We have selected the suitable Neural Network structure (number of best-hidden layers and activation function classes) to construct the e-health system. In addition, the e-health system relied on data from doctors to understand the Neural Network. In the validation method, the total evaluation of the proposed healthcare system for diagnostics provides dependability under various patient conditions. Based on evaluation and simulation findings, a dual hidden layer of feed-forward NN and its neurons store the tanh function more effectively than other NN. To overcome challenges, this study will integrate artificial intelligence with IoT. This study aims to determine the NN's optimal layer counts and activation function variations.

Introduction

Deep Learning allows AI to mimic how a human brain's neural network works. It is related to the field of Machine Learning. The data input method is the main variation between machine learning and deep Learning. Machine learning uses structured and unstructured data to learn from it and predict the future. On the other hand, deep learning networks use many layers of deep neural networks to make predictions about the future. There are input, hidden, and output layers in a neural network (s). It can determine the data's patterns, noise, and complicated parts. When the relationships between the features are linear, neural networks are rarely the best choice. Instead, machine learning can be used. Even when an artificial neural network networks for linear correlation, there is no requirement for a hidden layer. Neural networks work out the weighted sums. The weighted sum is transmitted as an input to the activation function in the hidden layers. The activation function connects the inputs with the target values. It utilizes the weighted total of input as input to the function, applies a bias, and determines whether the neuron should activate afterward. The expected output is shown in the output layer, and the produced models are compared to the real result. After the NN has been trained, the model improves the network using the backpropagation process. The error rate decreases based on the cost function.

Input and output layer neuron counts

The biased input layer's number of neurons can often match the importance of the features in the data. There will generally be one supplementary input layer for bias. If the model is used as a regression model or a classifier, then the number of neurons in the outputs will differ. The output layer of a regressor model will consist of a single neuron. In contrast, the output layer of a classifier model may include anywhere from a single neuron to several neurons, depending on the model's class label. The optimum number of neurons to deploy in the hidden layers may be calculated using various rule-of-thumb approaches, such as the ones listed below. To achieve the best possible outcomes, ensure that the hidden layer's sizer is in the middle of the sizes of the input and output layers. Additionally, ensure that the overall number of neurons in the hidden layer does not
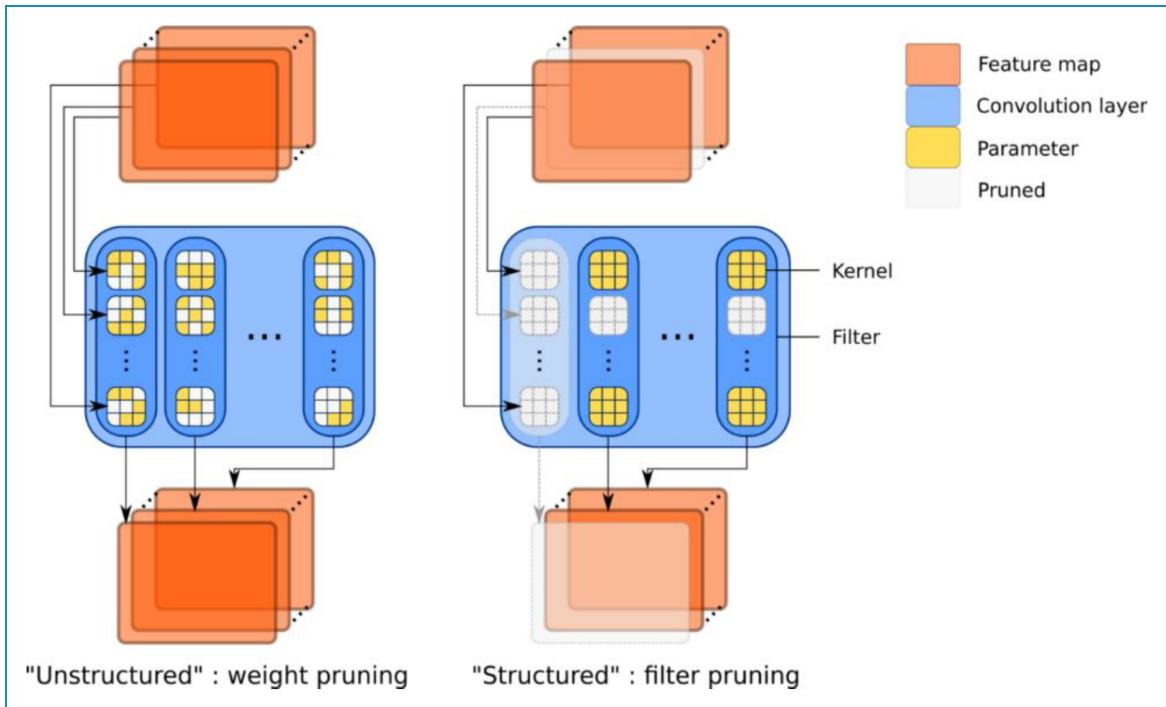
Figure 1. Depicts a pruning procedure.

exceed two-thirds the length and width of the input layer plus the dimensions of the output nodes.

The number of neurons and layers required for the hidden layer are also affected by the training examples, the number of misfits, the complexity of the data that must be learned, and the kind of activation functions utilized. The average number of neurons in the input and output layers may be used to determine the number of neurons in the hidden layer, which should be appropriate for solving most issues. Insufficient statistical power and underfitting will result from using a smaller-than-optimal number of neurons. Picking too many neurons might lengthen training time, increase variation, and risk overfitting.

Pruning

Optimizing the hidden neuron count by pruning improves computational and resolution speeds. Determining which neurons aren't contributing to the network's performance reduces the network's size during training. The neuronal weights may also be examined to get the diagnosis. Weights close to 0 are less significant. Such nodes are cut off during a pruning session. Figure 1 depicts a pruning procedure.

Artificial neural network ANN

This concept is mimicked by artificial neural networks in which our model is represented by neurons connected through edges (similar to axons). A neuron's value is simply
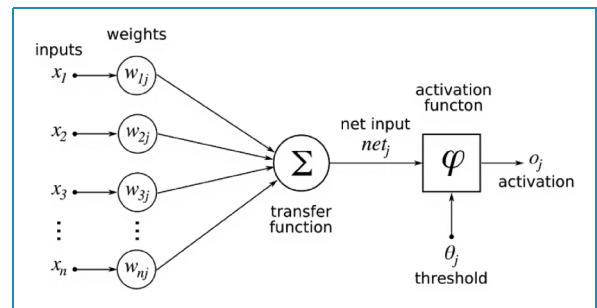

Figure 2. Demonstrates the activation function model.

the total of the value of preceding neurons connecting to it, weighted by the sizes of their edges. The neuron is then sent via an activation function, which determines how much it should be activated. Figure 2 demonstrates the activation function model.

ANNs are only an embellished form of matrix multiplication. Each layer of an ANN is represented by a vector, whereas matrices represent the weights between layers. Formally, it refers to them as tensors since their measures may change. This research aims to identify the optimal configurations of neural network hidden layers and nonlinear activation types. (NN). Provide some background on the patient information sent via the Internet of Things (IoT) protocols. Medical sensor data is processed by NN, which then draws the correct conclusion about the patient's condition. After that, the report is delivered to the attending physician.
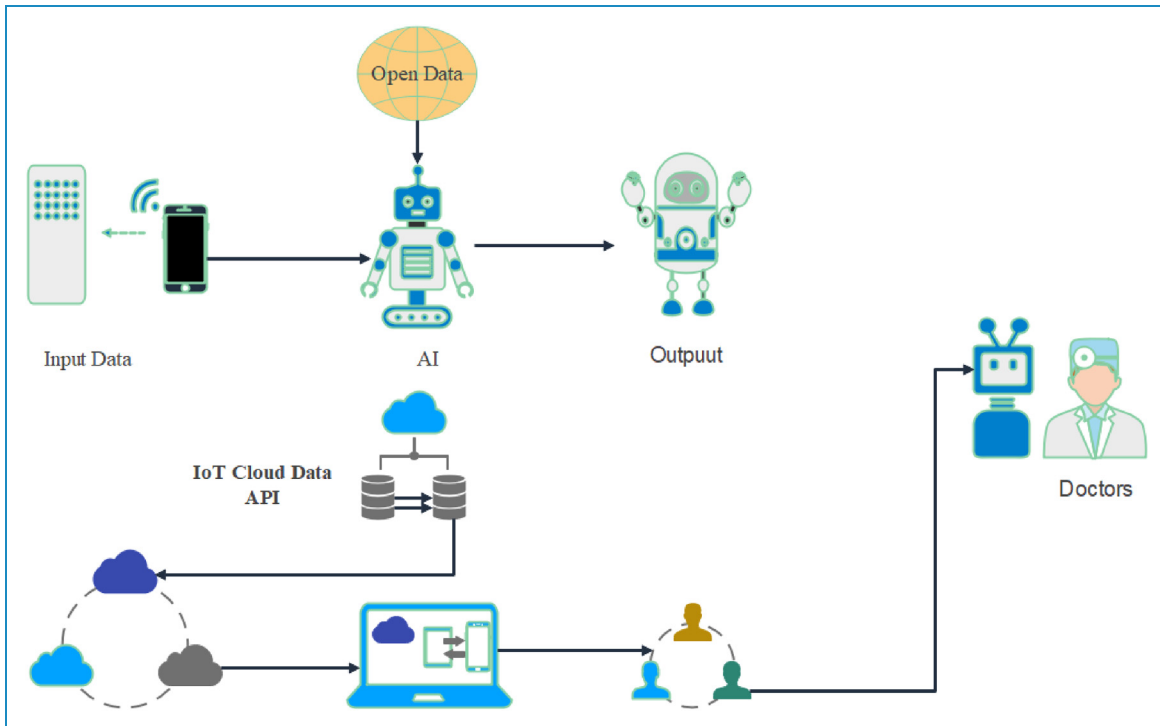
Figure 3. Display the combination of AI and IoT model.

Using the proposed solution, individuals may automatically identify and predict disease and assist physicians in detecting and analyzing the illness remotely without visiting the hospital. Combined with the IoT. A combination of AI and IoT is presented in Figure 3.

Three layers compensate for our structure: the input, hidden, and output layers. The input layer, the 16-dimensional feature vector of the picture, helps simplify the process. To represent a more generalized version of the input features, the hidden layer uses a four-dimensional vector of neurons. Multiplying the input vector by the 16-by-4 weights matrix W1 creates the hidden layer. Such as the hidden layer, and the output layer is generated by multiplying the input layer by a weight matrix.

Deep neural networks

These ANNs can become Deep Neural Networks by adding as many hidden layers as we require to get the desired depth (DNN). ANN versus DNN model is presented in Figure 4.

As a starting point for training a neural network, we use random values for the weights to exaggerate things. From the input layer, where our predictions are stored, we go to the final output layer. The weight matrices are modified significantly based on the prediction error we compute. Repetition continues until the weight remains the same. That needs to meet the requirements of the initial gradient descent and backpropagation algorithms, but it's good

enough for using neural networks. We can see the reduction in error (loss) as a function of changing the weights in this interactive Version.

Activation function for a deep neural network

Without a doubt, activation functions play a vital role in stimulating the activity of the latent nodes in neural networks and deep learning to provide better results. Non-linearity is introduced into the model through the activation function to achieve the goal at issue. If a node in an artificial neural network receives a certain. Input or Set of inputs, the output of that node is determined by its activation function. An embedded system's input may be considered nodes in a digital network of activation functions. If the input is positive, the rectified linear activation function (ReLU) will provide that data value; otherwise, it will return zero. It is now the default algorithm for many neural network types because of its accessibility in TRA and ability to provide satisfying performance. Leaky ReLU, ELU, SiLU, etc., are additional ReLU versions that improve performance on certain tasks.

There was a period when sigmoid and tanh activation functions were more common since they were simple, differentiable, and widely used. Nonetheless, these functions eventually saturate, making vanishing gradients a more pressing issue. The Rectified Linear Unit (ReLU) is the most widely used activation function for resolving this
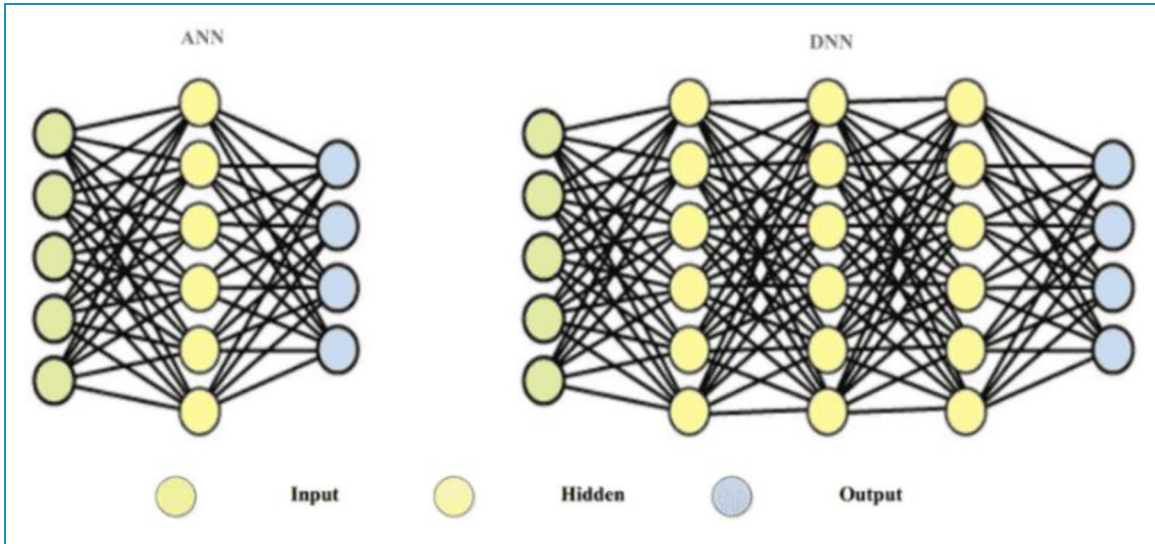
Figure 4. Display ANN versus DNN model.

issue. An activation function in a neural network is a function that takes a weighted sum of inputs from nodes and returns an activation for the nodes or outputs for that input. Activation functions like the logistic sigmoid (influenced by probability and statistics and regression models) and its more useful Version, hyperbolic tangent, had been frequently employed until 2011, when it was shown that a new activation function might support the advanced skills of deeper networks.

ReLUs

Rössig and Petkovic 2021, present an overview of algorithmic techniques that reestablish logical assurances on the behavior of neural networks, and the authors emphasize how these approaches operate. In addition, the authors give some new theoretical findings regarding the approximations of ReLU neural networks.[1] Jia and Li 2017 Rectifiers are deep neural networks' most common activation function. When the rectifier is used, the result is referred to as a rectified linear unit (ReLU).[2] While being one of the top activation functions, ReLU has yet to see rapid adoption for a number of reasons. That was the fact because it was not distinguishable at the 0 points. Differentiable functions like sigmoid and tanh were often used by scholars. However, it has been determined that ReLU is the ideal algorithm for deep Learning because of its simplicity and convenience. There is no non-differentiable point in the activation function of ReLU

outside of 0. If the function's value is larger than 0, we simply check at that value. Therefore, the following operation may be used:

In other words, $f(x) = \max 0, z$

If the input is more than zero, then
if the input is zero, return 0; if not, return input.
A negative number is automatically converted to zero, while positive numbers are increased to their highest possible value.

Backpropagation in a neural network

Training a neural network requires backpropagation, which means sending error rates back through the network to improve accuracy. The core of training a neural network is the backpropagation algorithm. An epoch-by-epoch method of optimizing a neural network's weights using information about its error rate (called loss) from the prior iteration (i.e., iteration). Lower error rates and improved generalization are two benefits of fine-tuning the weights. Differencing the ReLU for calculating backpropagation in neural networks is straightforward. Therefore, only one simplification is performed, assuming that the variant at the origin, 0, is likewise 0. If we take the variation of a function, then we get a number representing that function's slope. If the value is less than zero, the slope is zero; if it's more than zero, it's one.

	$f(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases} = \max\{0, x\} = x\mathbf{1}_{x>0}$	$f'(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ 1 & \text{for } x > 0 \end{cases}$	$[0, \infty)$

Advancements of the ReLU activation function

The following are the primary benefits that come with using the ReLU activation function:

1. The most often used activation functions for training convolutional layers and deep learning models are deep Learning and convolutional layers.
2. Simplicity of Computation: The rectifier function may be implemented with the minimum effort since all required is the max() function.
3. Truly representative Sparsity: The idea that the rectifier function can produce a zero number is an essential advantage of using this function.
4. Linear Behavior: It is much simpler to optimize a neural network when the behavior of the network is linear or very close to linear.

However, the Rectified Linear Unit automatically converts negative values to zero. This makes it more difficult for the model to fit the data or train itself to use it adequately. The ReLU activation function immediately converts any negative input into zero, which significantly impacts the final graph since it does not accurately map the negative values. Fixing this problem is as simple as using various variations of the ReLU activation function, such as the Leaky ReLU or one of the other versions.

Activation functions of the sigmoid and tanh

The neural network is composed of several layers of nodes, and it is programmed to learn how to convert different inputs to outputs depending on the different inputs that are given to it. In the case of a specified point, the inputs are divided by the weights contained in that node, and the final result is added to the total weights for that node. This number may be the activation of all the network nodes that have been added together. Applying an activation function to the cumulative activation may convert the summed activation into the particular output or activation of the node. This is achieved with the use of an activation function.

Because no transform is applied to the function, it is known as a linear activation function because it is the simplest activation function. It is very simple to train a network with linear activation functions; however, such a network cannot train complex mapping functions since it lacks the required information. Even at the output layer of a network designed to predict a quantity (for example, a regression issue), linear activation functions were always employed. If nodes utilize nonlinear activation functions, they will be able to learn more complicated structures in the data since these functions make it possible for them to grasp more extensive structures in the data. The sigmoid and rectified activation functions are two of the most common nonlinear activation functions. Both of these nonlinear activation functions are nonlinear.

When it comes to neural network models, the sigmoid activation function, sometimes known as the logistic function, has been considered one of the most prominent activation functions. The amount of the data transmitted to the function causes a transformation of that data into a value that falls somewhere between 0.0 and 1.0. When an input value is significantly less than 0.0, it is snapped to 1.0, and if an input value is much greater than 1.0, it is converted to the value 1.0. In the same way, if an input value is much greater than 1.0, it is changed to the value 1.0. Addressing the form of the function, the curves are S-shaped and range from 0 to 0.5 to 1 for all the different inputs that may be used. This range is from 0 to 1.0.

The hyperbolic tangent function, sometimes known as tanh for short, is a similarly shaped nonlinear activation function that produces values that range from −1.0 to 1.0. The function's name indicates this. The Sigmoid and tanh functions and many other mathematical operations are affected by the more general issue of saturating at a given point in their calculation. This implies that big values will be rounded up to 1.0 for the tanh and sigmoid functions, while small values will be rounded down to −1 or 0. Because of this, these functions are only sensitive to changes in the midpoints of their inputs, such as 0.5 for the sigmoid function and 0.0 for the tanh function.

It is important to remember that the function will still reach saturation and have a finite sensitivity independent of whether or not the accumulation of activation is included inside the provided input node. After the learning algorithm has reached capacity, it will be tough to continue adjusting the weights to increase the model's performance. Once the learning algorithm has reached capacity, it will become challenging to adapt the weights. In summary, even if the hardware capabilities of GPUs continued to improve, extremely deep neural networks that used sigmoid or tanh activation functions were more challenging to train than before because of the GPU's restricted capabilities.

When these nonlinear activation functions are used in big networks, gradient information is not sent to deep layers inside the network. If there is an error, it will make its way back through the network, and as a direct outcome of this, the network's weights will be modified. The error rate transmitted through each new layer that is propagated substantially lowers with each subsequent layer provided the derivatives of the selected activation function. The inability of deep (multilayered) neural networks to train properly is a consequence of this issue, referred to as the vanishing gradient problem. The application of activation functions enables neural networks to learn complicated mapping functions without a doubt, yet deep learning techniques cannot use these functions in any way.

Privacy and security challenges in deep learning decision support

The IoT and deep learning in Healthcare bring about promising advancements in decision support systems. However, this amalgamation also raises critical challenges related to privacy and security that must be carefully addressed to ensure the confidentiality, integrity, and compliance of sensitive patient information.

One of the primary concerns revolves around collecting, transmitting, and analyzing health data through IoT devices. The interconnected nature of these devices introduces vulnerabilities that malicious actors could exploit. Robust encryption protocols become imperative to secure data during transit and storage. Without adequate protection, patient information becomes susceptible to interception, tampering, or unauthorized access, posing serious risks to individual privacy and the overall integrity of healthcare systems.

Authentication and authorization mechanisms play a pivotal role in mitigating security risks. Strict controls must be in place to prevent unauthorized access to our IoT devices and deep learning systems. Unauthorized access could not only compromise patient data but also potentially manipulate the functioning of the decision support system, leading to incorrect diagnoses or treatment recommendations. Implementing multifactor authentication and regularly updating access privileges are essential measures to bolster security.

In continuous data collection, privacy concerns come to the forefront. The constant stream of health data from IoT devices necessitates adherence to stringent regulations, such as the Health Insurance Portability and Accountability Act (HIPAA). Compliance with these regulations is not only a legal requirement but also crucial for maintaining the trust of patients and stakeholders in the healthcare ecosystem. Furthermore, implementing anonymization techniques becomes imperative to protect the identities of individuals while still enabling meaningful analysis of health data. Striking the right balance between data utility and privacy is a delicate but essential aspect of deploying deep learning decision support systems in Healthcare.

As the research aims to integrate artificial intelligence with IoT, it must stay vigilant in addressing the evolving landscape of security threats. This entails continuous updates to device firmware to patch vulnerabilities and the implementation of secure communication protocols. The healthcare technology field is dynamic, and staying ahead of emerging threats is vital to safeguard patient information effectively. Additionally, compliance with legal frameworks beyond HIPAA, including regional and international data protection laws, is crucial to ensure a comprehensive approach to data security.

In conclusion, integrating IoT and deep learning in healthcare decision support systems holds great potential, but it comes with significant privacy and security challenges. Encryption, authentication, and authorization measures project data integrity and unauthorized access. Adherence to regulations like HIPAA, along with the implementation of anonymization techniques, addresses privacy concerns. Continuous efforts to update firmware, employ secure communication protocols, and stay abreast of evolving security threats are essential to secure patient information in the rapidly advancing landscape of healthcare technology. As the field progresses, a holistic and proactive approach to privacy and security will be paramount for the successful and ethical implementation of these transformative technologies in Healthcare.

Motivation

The integration of deep neural networks (DNNs) and IoT technology in Healthcare holds significant motivation for several reasons:

Advanced-data analysis. DNNs excel at learning from complex and unstructured data. In Healthcare, a vast amount of data is generated from various sources such as medical records, patient monitoring devices, genomics, and wearable sensors. By leveraging IoT devices to collect and transmit this data, DNNs can analyze it to extract valuable insights, patterns, and correlations. This enables healthcare professionals to make more accurate diagnoses, predict patient outcomes, and develop personalized treatment plans.

Real-time monitoring and intervention. IoT devices in Healthcare, including wearables and remote monitoring sensors, can continuously gather patient data such as vital signs, activity levels, and medication adherence. Integrating DNNs with IoT allows this real-time data to integrating DNNs with IoT allows this real-time data to be processed and analyzed. Healthcare providers can receive alerts or notifications if anomalies or critical situations are detected, allowing for immediate interventions and timely patient care. DNNs combined with IoT can contribute to efficient resource allocation in healthcare settings. For chronic conditions such as diabetes, asthma, or cardiovascular diseases, IoT devices can collect patient data, including glucose levels, respiratory parameters, or blood pressure, and feed it into DNN models. These models can provide personalized insights and recommendations for disease management, medication adjustments, or lifestyle modifications. This helps patients and healthcare providers optimize treatment plans and achieve better health outcomes. IoT devices enable patients to participate in their own Healthcare. Connected devices, mobile apps, and smart wearables can provide real-time feedback, reminders, and educational content to promote healthier habits and adherence to treatment plans. By integrating DNNs, these IoT devices can offer personalized recommendations

based on the individual's health data, empowering patients to take proactive steps in managing their health and well-being. Overall, the integration of DNNs and IoT technology in Healthcare can lead to more accurate diagnoses, personalized treatment plans, proactive monitoring, improved patient engagement, efficient resource utilization, and better public health management. However, it is crucial to address privacy, security, and ethical considerations to ensure the responsible use of data and maintain patient confidentiality. The main contribution of this research is to determine the optimal number of hidden layers for a neural network and explore different variations of activation functions.

Additionally, the study aims to analyze various frameworks that can be used to compare and create fast neural networks. The ultimate goal is to investigate innovative techniques that accelerate the training process of neural networks while maintaining high accuracy. Our research aims to address several gaps in the current understanding and practice of training neural networks.

Optimal number of hidden layers. Determining the ideal number of hidden layers in a neural network is crucial but challenging. There is a lack of consensus on the optimal depth of neural networks for different applications. Our research aims to fill this gap by exploring the relationship between the number of hidden layers and network performance, specifically focusing on achieving higher accuracy.

Activation function variations. Activation functions play a vital role in introducing non-linearity to neural networks. While commonly used activation functions like sigmoid or ReLU have been extensively studied, alternative functions may offer better performance in terms of accuracy and training speed. Our research aims to investigate different activation function variations to identify potential improvements in training efficiency.

Framework comparison for speeding up training. With the increasing complexity and size of neural networks, there is a need for efficient frameworks that can accelerate the training process. Our study aims to analyze and compare various frameworks, likely including well-known ones like TensorFlow or PyTorch, to identify the most effective approaches for speeding up neural network training without compromising accuracy.

Innovative techniques for speed-accuracy trade-offs. As training deep neural networks can be computationally expensive, finding innovative techniques that can reduce training time while preserving or even improving accuracy is a significant research challenge. Our research aims to explore and propose novel methods to accelerate training without sacrificing accuracy, potentially leveraging
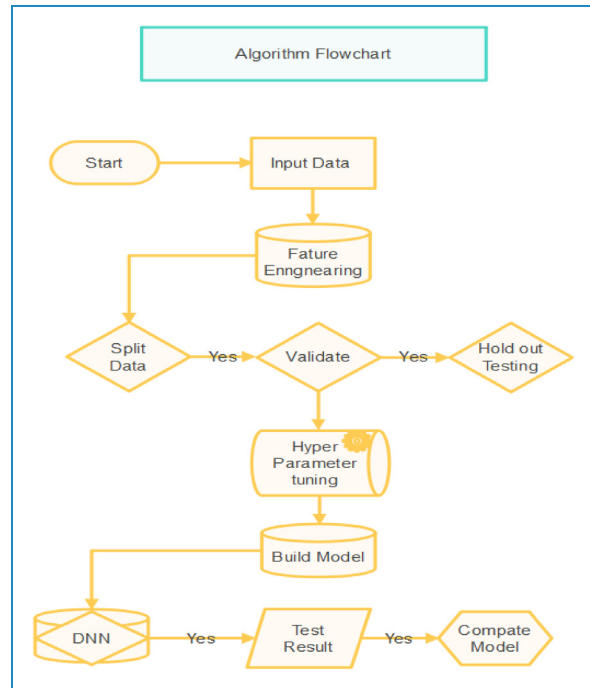

Figure 5. Display the algorithm flow chart.

advancements in areas like network architecture, optimization algorithms, or distributed computing.

Overall, the main contributions of the current research work are: By addressing these gaps, the research seeks to contribute to the development of faster and more accurate neural networks, enabling practical applications in various domains where computational efficiency is critical, such as real-time analysis of medical data, autonomous systems, or large-scale data processing.

This paper aims to present a novel approach that combines deep neural networks (DNN) with the IoT technology for healthcare applications. In Section II, we review relevant literature on DNN and its applications in Healthcare. Section III introduces our proposed methodology, outlining the architecture and data flow of the DNN model integrated with IoT devices. Section IV presents the results and discussion of our experiments, demonstrating the effectiveness of the proposed approach. Finally, in Section V, we conclude the paper by discussing the limitations of our study and suggesting avenues for future research. The procedure for processing is provided in Figure 5.

Literature review

Deep Learning is the most commonly debated topic in machine learning. It has a wide range of applications, including machine learning, language translation, language processing, graphic object recognition, diagnosis of illnesses, clinical trials, bioinformatics, biomedicine, and a number of others. Among these applications, the ones

related to the fields of medicine and Healthcare are showing significant growth. The rise in popularity of deep learning may be due to a variety of variables, the most significant of which is the spread of massive data, the IoT, linked devices, and high-performance processors with GPUs and TPUs. The Internet of Medical Things (IoT), digital images, data from electronic medical records (EMRs), genetic information, and centralized medical databases are some of the most important data sources for deep learning applications. The most important aspects of deep Learning are privacy, the optimization of the quality of service, and deployment, according to a number of potential issues. These aspects include security, quality of service optimization, and deployment. In Bolhasani et al., based on the Comprehensive Literature Review, this study evaluates deep convolutional neural networks for IoT systems in health care. This paper examines the studies completed between 2010 and 2020, selected from 44 previously published research publications.[3]

In Malarvizhi et al., the IoT-enabled e-healthcare apps significantly contribute to society by enabling healthcare monitoring services in an intelligent environment. Due to the large number of users and the availability of their private data in this age of high-speed Internet and cloud databases, the security of the healthcare system must be considered a significant issue. Issues about the privacy and security of patient data are raised by the need to securely maintain electronic versions of patient health records. In addition, managing large volumes of data with standard classifiers is a reasonably complicated task nowadays. Numerous deep learning methods are available for effectively categorizing the enormous amount of data. The authors present a novel health monitoring system that tracks the disease status by predicting illnesses based on the original data from distant patients.[4]

In Wassan et al., the convolutional neural network model architecture and conv1d algorithm were used to forecast frost events.[5] Shaikh et al. introduced a MANET multicast protocol that is both lightweight and easy to use.[6] In Wassan et al., model's efficacy has been validated. Experiments in the real world validate the effectiveness of our suggested algorithm in preserving user privacy.[7]

In Zikria et al., integrating deep learning methods into IoT to improve application performance is challenging. Next-generation IoTs networks need a combination of these methods to balance computational costs and efficiency. To meet the requirements of machine learning, the IoTs, and seamless integration, a complete redo of the communication infrastructure from the network's physical layer to the application level is required. Therefore, applications built on the improved stack will, and the network will be simpler to deploy widely.[8] Wassan, et al., presented Machine Learning Methods Applied to an Analysis of Public Perceptions.[9]

In Fahmy et al., during epidemics, such as those that result from the transmission of a virus, it could be extremely difficult to get people to come to the hospital for treatment (COVID-19). Artificial intelligence (AI) and the IoT solutions will be used in this study to investigate and address this issue.[10]

Deep learning (DL) algorithms are used in the diagnosis of diseases. In the current study by Ragab et al., using the best stacked sparse denoising autoencoder (SSDA) technique, i.e., OSSDA, the authors propose a unique approach to IoT-based physical health monitoring and control. The proposed model collects patient data via a collection of IoT devices. During disease diagnosis, the imbalanced class situation poses significant challenges.[11] In Ullah et al., by incorporating the notion of innovation dissemination, this research offered a revised version of the Technology Acceptance Model.[12]

Iranpak et al. use a prioritization method to prioritize sensitive data in the IoT. In cloud applications, an LSTM deep neural network is applied to categorize and remotely monitor patients' conditions, a significant innovation.[13]

Ganesh et al., study presented Different CNNs show various semantic features of the picture depending on their deeper architectures; hence, a collection of CNN architectures enables it feasible to extract the elements with higher quality than traditional methods.[14] Vasan et al., study provide an effective malware threat detection model for IoT across architectures utilizing advanced ensemble learning (MTHAEL).[15] Previously, Wassan et al. presented a machine learning and deep learning model on sentimental analysis.[16] Vasan et al., presented using a CNN-based deep learning architecture, IMCFN was developed as a revolutionary classifier to better detect malware by identifying variations among families of malware.[17]

Humayun et al. study suggest a CET (cutting-edge technologies)-a based approach called the (RTSHPMP) for ensuring the safety and privacy of travelers in addition to providing medical and legal support.[18] Muzammal et al. study presents SM Trust, a conceptual IoT routing protocol security approach based on mobility-based trust metrics.[19] Namasudra presented the negative consequences of lack of sleep are mitigated by the advantages of sleep described by Ref. 20. Wassan et al. employed machine learning and deep learning algorithms to examine e-commerce and e-banking datasets and discovered the top techniques on performance metrics.[21] Dietz work focuses on using deep Learning in experimental bioinformatics, medical imaging, extensive sensors, healthcare analytics, and public health.[22]

With this research, Harerimana et al., aimed to provide a more human-level justification for the potential applications of deep Learning with E-health. Authors provide technological ideas and blue-blueprints on how a deep learning algorithm can address each clinical activity and then discuss how these approaches can be used by health informatics experts.[23] In this work, Varshney & Singh propose a new method for generalizing the ReLU activation function

by use of a number of configurable slope parameters.[24] Placzek & Placzek, many weight coefficients in a neural network's internal layer matrices with a ReLU activation function represent zero during practical activities.

This phenomenon reduces the efficiency of the learning algorithm's progress. When attempting to analyze and describe the structure of an ANN, the number of layers in the ANN is often the first parameter presented. The learning mechanism of an ANN is hierarchical; therefore, the network is divided into smaller networks. The learning error is minimized by having each sub-network discover the superior value of its weight coefficients to use a local target function. The global target function minimum is found by coordinating the local solutions at the second collaboration level of the learning method. Every time a cycle completes, the coordinator must update the initial level of subnets with new coordination settings. The local processes operate and determine their weight coefficients utilizing the input and the training vectors. The feedback error is computed and transmitted to the coordinator at this step. This will continue until the total target functions are minimized by Ref. 25.

Activation functions are used in deep learning algorithms to process the network's inputs and produce the matching output. Deep learning models are helpful for image analysis, natural language processing, object classification, and prediction; Kiliçarslan & Celik are especially beneficial for evaluating big data with several parameters and predictions. Deep learning models often use conventional activation functions like sigmoid and tangent. On the other hand, the sigmoid and tangent-activated functions have the vanishing gradient issue.[26]

Zhang et al., provide a recurrent neural network (RNN)[27]-based NMPC technique for single-link flexible-joint (FJ) robot localization using multi-objective evolutionary optimization (DEO).[28] For real-time applications, Alrawashdeh and Purdy propose the Adaptive Linear Function (ALF), a rapid activation function that improves both the converging speed and precision of the deep learning algorithm.[29] In this article, Boullé et al., focus on neural networks with logical activation functions. In deep learning architectures, the efficiency of a neural network is deeply impacted by the activation function used. The authors show that rational artificial neural networks have optimum limitations in terms of complexity, and we demonstrate these bounds empirically.

The proposed method uses Deep Neural Networks (DNNs) and IoT technology. Here, we presented a detailed description of the superiority of the proposed method over existing literature and why there is a need for it.

Increased model complexity. With their multiple layers, Deep Neural Networks allow for increased model complexity compared to traditional shallow networks. This increased depth enables the model to learn more intricate and abstract representations from IoT-generated data, such as images. By leveraging deep learning techniques, the proposed method can capture complex patterns and relationships in the data, leading to improved accuracy and performance. One significant advantage of DNNs is their ability to automatically learn feature representations directly from raw data. Traditional methods often require manual feature engineering, where domain-specific experts design and extract relevant features.

In contrast, the proposed method based on DNNs eliminates the need for manual feature engineering, reducing human effort and potential biases. The network learns hierarchical representations of the data, enabling it to automatically extract meaningful features from IoT-generated images, resulting in more robust and discriminative representations. The proposed method based on DNNs excels in adaptability and generalization across diverse IoT data. The deep architecture can handle variations in image resolutions, sensor types, and environmental conditions. It learns to extract relevant features and patterns generalizable to unseen data. This adaptability is crucial in IoT applications where data from various sources with different characteristics need to be analyzed consistently and accurately.

Real-time processing. IoT generates vast data in real-time, requiring efficient and timely processing. DNNs can be optimized and deployed on high-performance hardware or specialized processors for real-time analysis. By leveraging the parallel computing capabilities of DNNs, the proposed method can handle the computational demands of processing IoT data in real-time, facilitating quick decision-making and response in time-sensitive applications. Scalability: With the rapid growth of IoT deployments, scalability becomes paramount. DNNs are inherently scalable and can handle large-scale datasets and deployments. The proposed method can efficiently process and analyze the ever-increasing volume of IoT-generated images. By leveraging distributed computing or parallel processing techniques, the method can scale horizontally to accommodate expanding IoT networks, making it suitable for applications spanning smart cities, industrial IoT, and healthcare systems. Edge computing brings computation closer to IoT devices, reducing latency, conserving network bandwidth, and enhancing privacy. Based on DNNs, the proposed method can be optimized and deployed on edge devices, enabling on-device processing and analysis of IoT-generated images. This capability eliminates the need for continuous data transmission to centralized servers, making it ideal for latency-sensitive applications and scenarios where data privacy is a concern. The proposed method based on Deep Neural Networks and IoT technology offers superiority over existing literature by leveraging increased model complexity, automatic feature extraction, adaptability to diverse data, real-time processing, scalability, and edge computing capability. Integrating DNNs and IoT technology addresses

the specific needs and challenges of analyzing IoT-generated image data, providing an advanced and efficient solution for various IoT applications.

Using IoT and deep Learning, the researchers showed a healthcare decision-support system. Using the suggested approach, we investigated how well deep Learning could be integrated with automated diagnosis and IoT features to provide more rapid data transmission over the web. To create the e-health system, we have selected the best NN structure number of best-hidden layers and activation function classes. In addition, the E-health system relied on data from doctors to understand the NN. In the validation method, the total evaluation of the proposed healthcare system for diagnostics provides dependability under various patient conditions. Based on evaluation and simulation findings, a dual hidden layer of feed-forward NN and its neurons store the tanh function more effectively than other NN. To overcome challenges, this study will integrate artificial intelligence with IoT. This study aims to determine the NN's optimal layer counts and activation function variations.

Table 1 summarizes the research on deep learning applications for IoT in Healthcare that has been done and evaluated.

The main contributions of the study can be summarized as follows:

The study proposes integrating deep learning algorithms with IoT technology in a healthcare decision support system. This integration enables the system to report patient data and deliver timely warnings to medical applications or electronic health information in case of patient health changes. The proposed system utilizes deep learning technology for automatic diagnosis. By analyzing patient data, deep learning algorithms can provide accurate and efficient diagnostic results, helping healthcare professionals make informed patient care decisions. The study focuses on selecting the suitable neural network structure, including the optimal number of hidden layers and activation function classes, for constructing the e-health system. This optimization ensures that the neural network performs effectively in processing and analyzing healthcare data. The proposed healthcare system undergoes validation to evaluate its performance and dependability under various patient conditions. This validation process ensures that the system provides reliable and accurate diagnostic results. Through evaluation and simulation, the study identifies that a dual hidden layer feed-forward neural network with neurons storing the tanh activation function performs more effectively compared to other configurations. This finding contributes to understanding the optimal neural network configuration for healthcare applications. The study recognizes the challenges in Healthcare and aims to overcome them by integrating artificial intelligence (AI) with IoT. This integration harnesses the power of AI algorithms, such as deep learning, to leverage the vast amount of data collected by IoT devices, enabling more efficient and effective healthcare solutions. Overall, the study's main contributions lie in the integration of deep learning and IoT for automatic diagnosis in Healthcare, optimization of the neural network structure, validation of the proposed system, identification of the optimal neural network configuration, and the vision of integrating AI with IoT to address healthcare challenges.

Table 1. Research conducted in deep learning uses for IoT in healthcare.

Reference	Main Topic	Publication Year
Zhao et al.[30]	MHMS	2019
Friday Nweke et al.[31]	human activity recognition	2018
Pasluosta et al.[32]	IoHT	2018
Alam et al.[33]	Data fusion	2017
Riazul Islam et al.[34]	IoT-based healthcare technologies	2015
S. Lakshmanaprabu et al.[35]	DNN classifier for the Identification of CKD	2019
Edmond S. L. Ho.[36]	DNN models on the cloud	2022
S. Wassan et al.[7]	(DPFL-HIoT) model	2022
Our Proposed Model	DNN HIOT	2023

Methodology

Our dataset is very small, so we may use fewer layers in the deep learning model. This will enable us to use our time. To perform the ReLU activation, we will make use of two layers that are completely connected. If the value being supplied is larger than zero, the ReLU activation will return 0, and else it will output the value being input directly. The dataset sample is shown in Figure 6. The time duration and place where the study was conducted are presented in Figure 7.

Class pandas core frame visualized in Table 2. The range index entries are 200 from 0 to 199, and the total data column is 16. There are three data types int64, bool, and float64.

Assumptions of our study

During the simulation phase of DNN research, we simplified the experimental setup and facilitated model training

and evaluation. The training and testing datasets are representative of real-world data distribution. While some scholars strive to collect diverse and comprehensive datasets, there might still be inherent biases or limitations in the data collection process. During data preprocessing, such as normalization, resizing, or cropping of input data. These preprocessing steps are classically performed to ensure consistency and compatibility across different samples. The provided labels or annotations for the training data are accurate. Simulation experiments have access to sufficient computational resources, such as high-performance GPUs, for training and evaluating DNN models. The availability of such resources can impact the scalability and efficiency of the proposed method when transitioning from simulation to real-world deployment. Our selected parameters, such as learning rate, batch size, and network architecture, are the

best for small data sets. These choices are based on empirical observations and prior knowledge. In some conditions, optimal hyperparameter selection may require additional experimentation and tuning.

Data analysis process

First, we transfer the dataset from our PC. CSV file provided.

Second Stage: Preprocessing, this operator contains all of the generic preprocessing processes.

Third, prepare your data ready for correlation analysis.

In the fourth stage, encoding, the data is one-hot encoded (because there is no target), and columns that contain too many nominal values are deleted.

Step 5: Remove Extraneous Data, such as Constant Columns

The 6th step is to collect a sample with no features and then further reduce the sample size based on the number of traits.

In the 7th stage, attributes are rearranged by organized alphabetically columns.

Correlation Matrix, the 8th Step: Construct a real connection

In the 9th step, you'll annotate and give a name to the output.

The building plan is shown in Figure 8.

Result

Performance of deep learning regression model

A common way to assess variability in statistics is the standard deviation (SD). It demonstrates how much

Figure 6. Presented the sample of the dataset.

Figure 7. Presented the data scoring history.

Table 2. Presented the data frame.

Column	Non-Null Count	Dtype
0 Row	200 non-null	int64
1 ID	200 non-null	int64
2 Fraud	200 non-null	bool
3 Prediction F	200 non-null	bool
4 Prediction T	200 non-null	float64
5 Conf	200 non-null	float64
6 Amount paid per doctor	200 non-null	int64
7 Amount paid per hospital	200 non-null	int64
8 Amount paid per year	200 non-null	int64
9 Amount paid per date	200 non-null	int64
10 Numbers prescription per doctor	200 non-null	int64
11 Numbers prescription per hospital	200 non-null	float64
12 Numbers prescription per year	200 non-null	float64
13 Numbers prescription per date	200 non-null	float64
14 Numbers	200 non-null	float64
15 Number prescribed per prescription	200 non-null	int64
16 Prediction fraud label	200 non-null	int64

variables differ from the mean (average). While a high SD shows that the data are dispersed throughout a wide range of values, a low SD indicates that the data points are near the mean. A static view of the Sample of data can be visualized in Figure 9.

Mean squared error (MSE). MSE measures the average squared difference between the predicted and actual values in the training set. It provides a measure of the overall training error of the model. A lower MSE indicates a better fit of the model to the training data, with 0 being the best possible value.

Root mean squared error (RMSE). RMSE is the square root of the MSE and represents the average magnitude of the prediction errors in the original scale of the target variable.

Similar to MSE, a lower RMSE indicates better model performance on the training data.

Coefficient of determination (R^2). R^2 measures the proportion of the variance in the dependent variable that the model on the training set can explain. It ranges from 0 to 1, where 1 indicates that the model explains all the variance in the data. A higher R^2 value signifies a better fit of the model to the training data.

Mean residual deviance. Mean Residual Deviance measures the average difference between the predicted and actual values, considering the model's goodness-of-fit on the training data. A lower value indicates a better fit of the model to the training data.

Mean absolute error (MAE). MAE calculates the average absolute difference between the predicted and actual values in the training set. It provides a measure of the average magnitude of the errors made by the model on the training data. A lower MAE indicates better model performance on the training set.

Root mean squared log error. Root Mean Squared Log Error is the square root of the mean of the logarithmic differences between the predicted and actual values in the training set. It is particularly useful when the target variable has a skewed distribution. Similar to RMSE, a lower value indicates better model performance on the training data.

These parameters help evaluate the performance and accuracy of the regression model on the training data. They provide perceptions of how well the model fits the training data and how closely the predicted values align with the actual values. It's important to consider these metrics collectively to assess the model's training performance quality.

We use multiple parameters to determine the most effective method to test how well our method works. In the next paragraph, we'll discuss how the calculation changes secret-shared Values. By adopting 19 train set features, we train our reliable model to predict healthcare cost's (numerical) target feature. We found that 0.89503 was the best choice because it gave us a good fit (R^2) and let us set enough coefficients to 0. To develop our stable model with this Set of parameters, we require 26 iterations. We use an R^2 of 0.89503, an MSE of 0.01094, an RMSE of 0.10458, a mean residual deviance of 0.01094, a mean absolute error of 0.07452, and a root mean squared log error of 0.07207. The regression Model training result is shown in Table 3.

After training the model on the train set, we applied the same parameters to the test set and obtained an R^2 of 0.90707, MSE of 0.01045, RMSE of 0.10224, mean residual deviation of 0.01045, MAE of 0.06954, and RMSE of 0.07051, validating our solution approach. The objective value of our secured model is higher than that
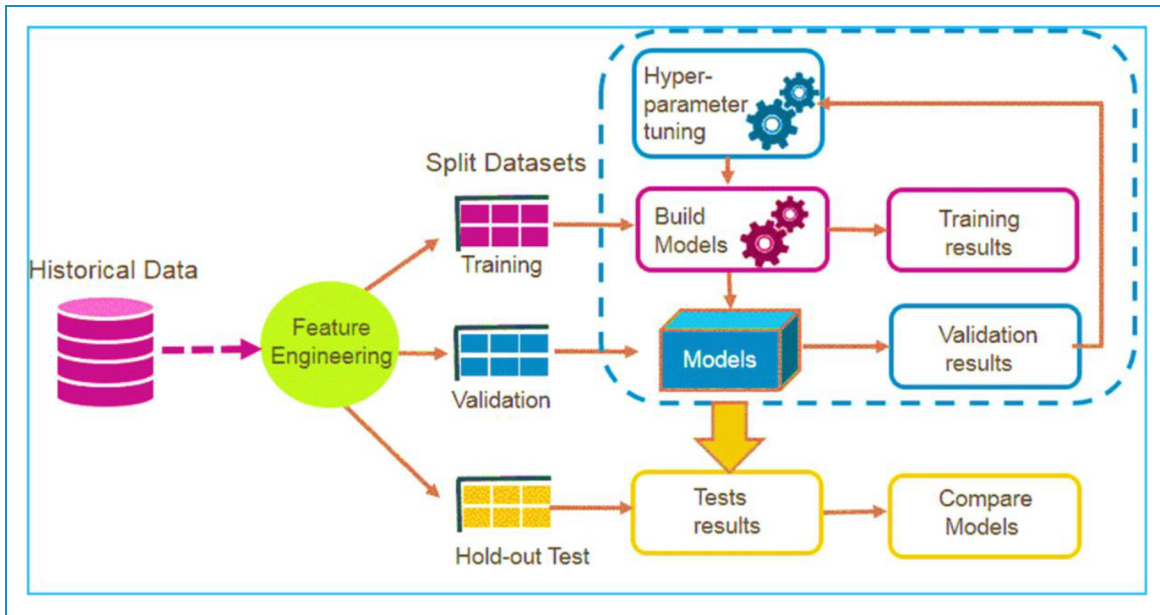
Figure 8. Display the model building plan.
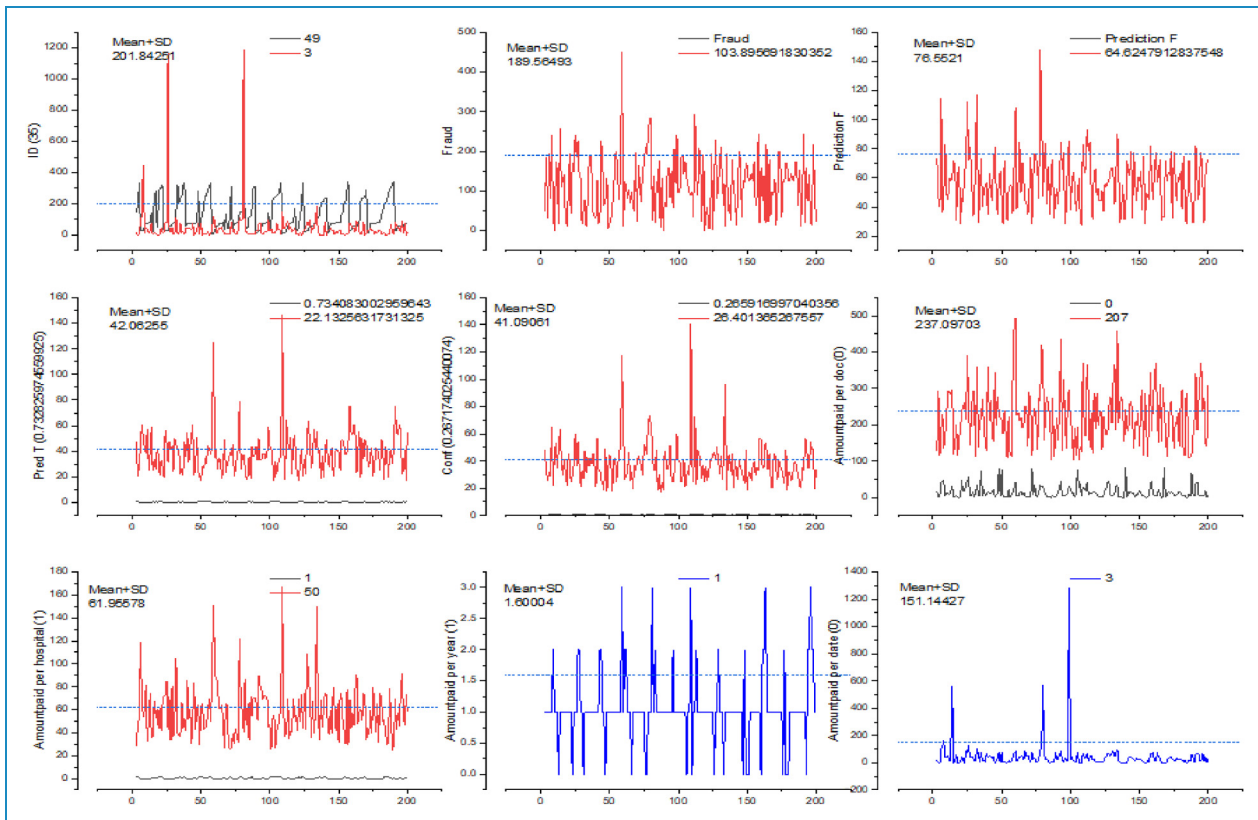

Figure 9. Presented the static view of the sample of data.

of the scikit-learn model, although the former performs better on goodness-of-fit criteria. As a result, our protected model performs quite well, marginally outperforming the (very optimized) scikit-learn model. Regression Table 4 displays the outcomes of the experimental results, and Figure 10 displays the confusion matrix.

Table 3. Presented the regression Model training result.

Regression Model Training Result	
MSE	0.01094
RMSE	0.10458
R^2	0.89503
Mean Residual Deviance	0.01094
Mean Absolute Error	0.07452
Root Mean Squared Log Error	0.07207

Table 4. Presented the regression model testing result.

Regression Model Testing Results	
MSE	0.01045
RMSE	0.10224
R^2	0.90707
Mean Residual Deviance	0.01045
Mean Absolute Error	0.06954
Root Mean Squared Log Error	0.07051

The standard deviation measures the degree of distribution of a collection of data. Each data point is compared to the average of all the data points, and the standard deviation gives a determined number that indicates whether the data points are clustered together or separated.

The performance results of our model indicate strong performance across various metrics. The Area Under the Curve (AUC) value 97.2 suggests excellent discrimination ability, particularly in binary classification tasks. The high accuracy of 98.2% indicates the overall correctness of predictions, while the low classification error of 30 points to a small proportion of misclassifications. The f-measure, a balanced metric between precision and recall, is at 97.9%. Precision (99.2%) reflects the model's ability to correctly identify positive instances, while recall (99.1%) measures its ability to capture all positive instances. Sensitivity (95.1%) indicates the model's effectiveness in identifying true positives, and specificity (96.2%) highlights its proficiency in correctly identifying true negatives. Overall, the results suggest a well-performing model with high accuracy and a balanced trade-off between precision and recall, demonstrating robustness in positive and negative instance

predictions. The result performance model table is presented in Table 5.

Table 6 provides models that demonstrate varying degrees of accuracy in their predictive capabilities. Baek and Chung's 2020 model achieved a relatively low accuracy of 0.85%, indicating potential limitations. Swetha's 2020 model stands out with an impressive accuracy of 97%, showcasing its robust predictive performance. Khanna, Selvaraj et al.'s 2023 model achieved a commendable 93% accuracy, while Wassan, Suhail et al.'s 2022 model lagged with an 82.5% accuracy. In contrast, the proposed model boasts the highest accuracy at 98.2%, surpassing all referenced models. This proposes that the proposed model exhibits exceptional predictive power, outperforming existing approaches and potentially offering a superior solution for a certain task.

The standard deviation of a normal distribution indicates how far values deviate from the mean. The static view of the fraud prediction, true and false, can be visualized in Figure 11.

Deep Learning develops artificial neural systems with several interconnected layers using a backpropagation algorithm and stochastic gradient descent. There may be hidden layers of neurons in the network that have the tanh, rectification, and max-out hyperparameters. Modern features like momentum training, dropout, active learning rate, rate annealed, and L1 or L2 regularization provide exceptional prediction performance. The worldwide model's parameters are multi-threadedly (asynchronously) trained on the data from that node, and the model-based data is then gradually augmented by model averaging over the entire network.

The method is executed on a single-node, direct H2O cluster initiated by the operator. The operation is parallel despite there just being a single node involved. The number of threads may be adjusted in the settings menu under Preferences and General. The optimal number of threads for the system is used automatically.

Successful predictions in the healthcare data sets are made using the H2O Deep Learning operator. There will be a classification done since its label is binomial. The Splitting Validation operator creates test and training datasets to evaluate the model. By default, the settings of the Deep Learning activator are used. To put it another way, we'll construct two hidden layers, each containing 50 neurons. The Accuracy measure is computed by linking the annotated Sample Set with a Performer (Binominal Classification) operator. Table 7 displays the Deep Learning Model, the labeled data, and the Performance Vector that resulted from the technique.

Regression using deep learning

The H2O Deep Learning function forecasts a dataset's numerical label attribute. Since its label is true, regression
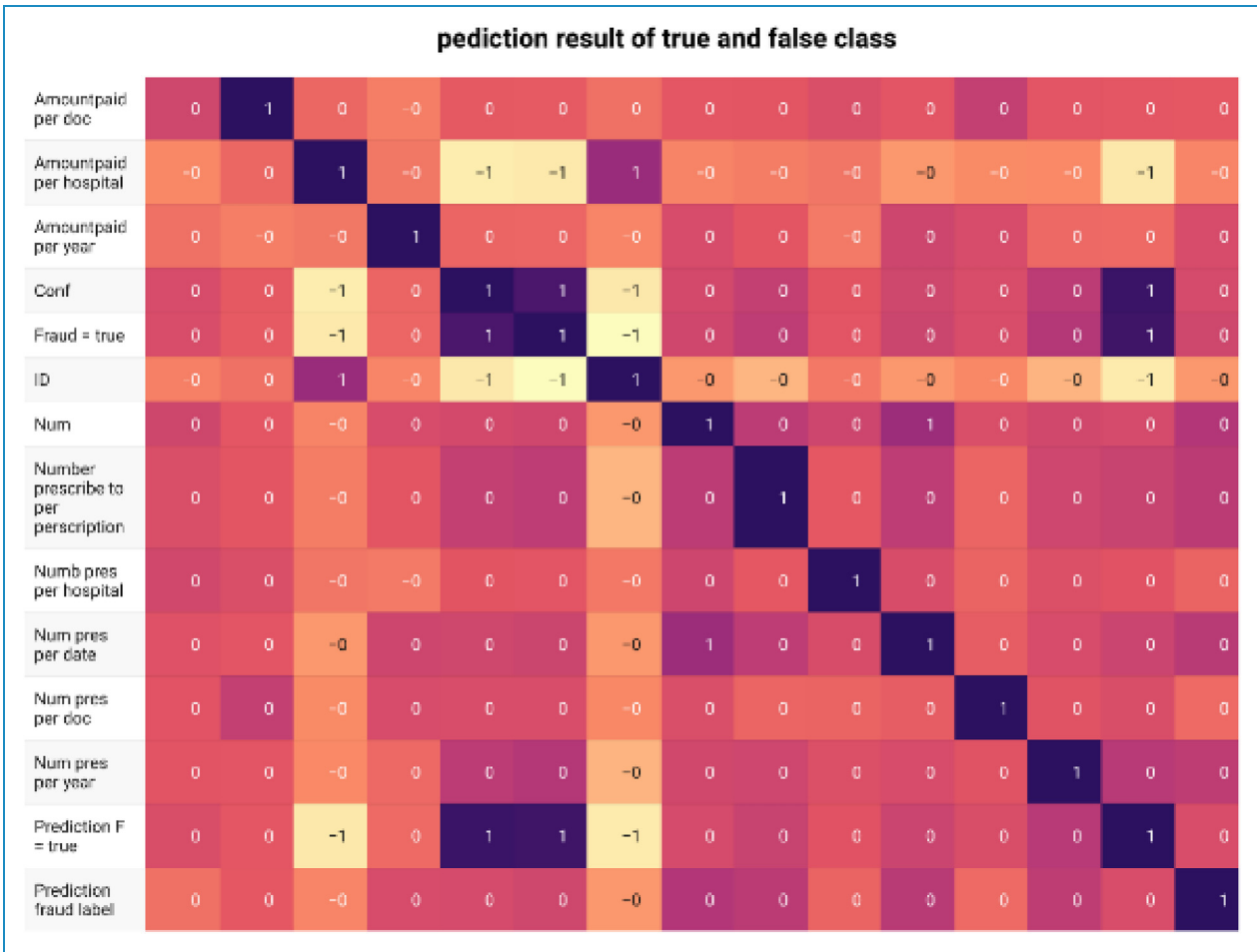
Figure 10. Displays the confusion matrix.

is performed automatically. The resulting data is divided into two pieces using the Split Data operator. The training data Set is the first output, and the evaluating data Set is the second. The Deep Learning controller (default) uses the adaptive learning rate. The program calculates the learning rate automatically based on the epsilon and rho variables. The sole non-default option is the hidden layer size, which uses three layers of 50 neurons each. After its application to the validation data, the labeled data is tied to the process performance.

ReLU in regression

Activation function (ReLU). We use activation functions on both hidden and output neurons to protect the neurons from falling too low or too high, which would work against the network's learning process. This prevents the neurons from falling too low or too high. Simply said, the arithmetic is more accurate when performed in this method, as shown in Figure 12.

Table 5. Presented the performance result.

Performance Result	
Criterion	Result
AUC	97.2
Accuracy	98.2
Classification error	30
f_measure	97.9
Precision	99.2
Recall	99.1
Sensitivity	95.1
Specificity	96.2

The activation function that is implemented in the output layer is the one that is considered to be the most effective. The expected result is continuous output when the NN is used to solve a regression issue. We continue to use the dataset of medical patients for the experiment. Rectified Linear Unit is one of the most straightforward and practical

activation functions. Thus, we employ it to ensure compliance with this guideline. Adam is only an improved form of the algorithm known as gradient descent, used for optimization. It is much quicker than several other optimizer techniques.

We trained our model for ten epochs, each consisting of a systematic run of the training data that is shown in Table 8. While Figure 13 displays the Classification result and iteration of the training result.

As we may iterate over batches of data, an epoch is not the same as a learning iteration. A new epoch is started. However, the model has completed an iteration on all the training data each time.

The RMSE is still under one variance, showing that the data are usually distributed. This reflects positively on the model's effectiveness.

Table 6. Presented comparing performance result in existence medical data model result.

Reference	Performance Result (Accuracy)
Baek and Chung 2020[37]	0.85%
Swetha 2020[38]	97%.
Khanna et al. 2023[39]	93%
Wassan et al. 2022[7]	82.5%
Our Proposed Model	**98.2%**

Discussion

The study described aims to investigate the ideal number of hidden layers for a neural network and explore different
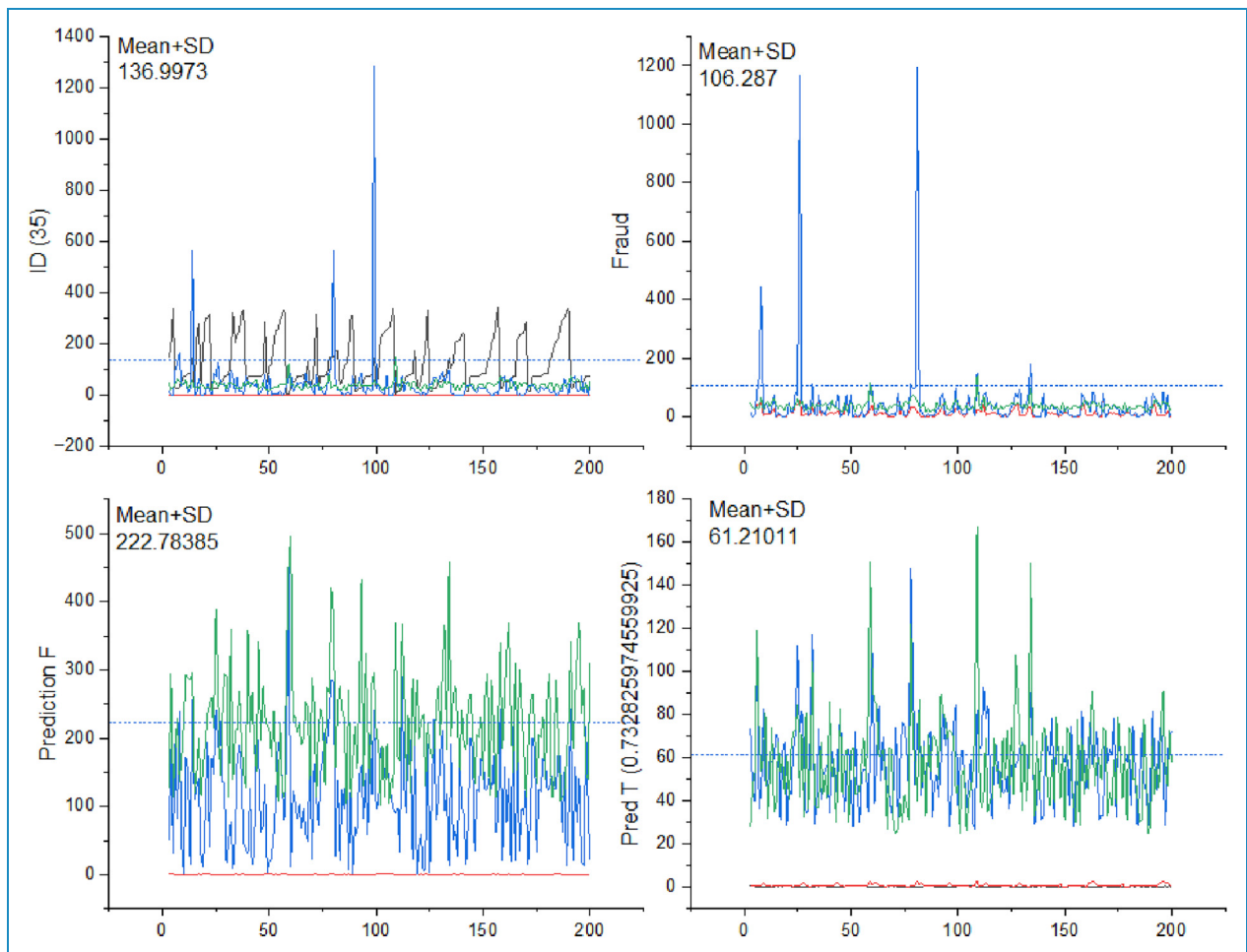

Figure 11. Provided the graphical view of fraud prediction of true and false value.

Table 7. Displays the deep learning model.

Status of Neuron Layers											
Layer Units	Type Dropout	L1		L2	Mean Rate	Rate RMS	Momentum	Mean Weight	Weight RMS	Mean Bias	Bias RMS
1	50 Rectifier	0 0.000010		0	0.204566	0.40263	0	0.001	0.16777	0.50172	0.02121
2	50 Rectifier	0 0.000010		0	0.009399	0.01716	0	−0.00019	0.13862	0.99929	0.02088
3	1 Linear	0.00001		0	0.000492	0.00021	0	0.02027	0.20515	−0.01071	0

variations of activation functions. Additionally, the study analyzes various frameworks to compare and enhance the speed of neural network training while maintaining accuracy. In terms of the methods, the study utilized a sample dataset from 200, which was collected from www.Kaggle.com. The total number of layers in the deep learning model was reduced to improve efficiency. The study employed the rectified linear unit (ReLU) activation function, utilizing two fully connected layers. The ReLU activation function returns 0 if the input value is larger than zero and outputs the input value directly otherwise. The discussion of the study would involve interpreting and analyzing the findings and their implications. This could include assessing the performance of different activation function variations and the impact of varying the number of hidden layers on the neural network's accuracy and training speed. The discussion may also involve comparing and contrasting the results obtained using different frameworks for training neural networks and their implications for real-world applications. Overall, the study's discussion would provide insights into the effectiveness of different techniques in optimizing neural network architectures, improving training speed, and maintaining accuracy to identify innovative approaches to accelerate the training process without sacrificing performance. The study described focuses on utilizing deep learning algorithms to develop a healthcare decision support system that integrates IoT features for real-time data transmission and self-diagnosis. The objective is to improve patient care by ensuring timely and effective treatment based on personalized data analysis. The proposed system leverages deep learning technology, specifically neural networks, to design the e-health system. The study investigates the selection of the optimal number of hidden layers and activation function variations in the neural network architecture. The evaluation and simulation findings suggest that a dual hidden layer feed-forward neural network with the tanh activation function provides more reliable results when applied to diagnostics. The integration of artificial intelligence (AI) with IoT is highlighted as a means to overcome challenges in the healthcare domain. By combining these technologies, the study aims to
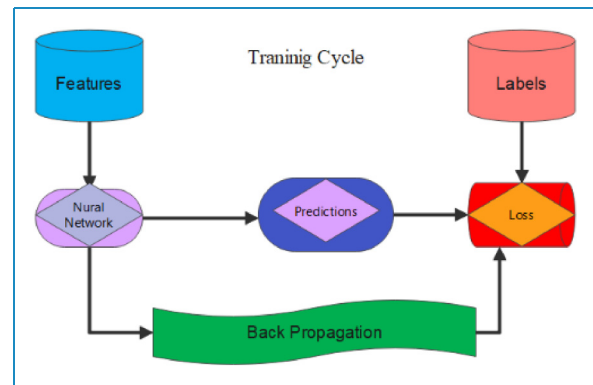

Figure 12. Present the training cycle of the deep learning model.

enhance the efficiency and effectiveness of the e-health system. The study addresses the growing need for advanced systems in Healthcare that can provide real-time monitoring and personalized care. By incorporating deep learning algorithms and IoT features, the proposed system offers the potential to improve patient outcomes and ensure timely interventions. Utilizing neural networks allows for complex data analysis and pattern recognition, which can aid in diagnosing patients accurately. Selecting the optimal number of hidden layers and activation functions is crucial in achieving reliable results and ensuring the system's effectiveness. By integrating AI with IoT, the study recognizes the potential to leverage the massive amount of data generated by IoT devices and apply advanced analytics for healthcare decision-making. This integration could improve efficiency in diagnosing and treating patients and provide timely warnings and alerts to healthcare providers. One limitation of the research could be the specific focus on finding the ideal number of hidden layers and activation function variations without considering other factors that may influence the neural network's performance, such as the choice of optimization algorithm or hyperparameter tuning. This study may have used a limited dataset or specific healthcare scenarios, which could affect the generalizability of the findings to different healthcare applications or datasets. The comparison of different frameworks for creating fast

Table 8. Displays the classification result and iteration of the training result.

Epochs Iterations	Samples	Training RMSE	Training Deviance	Training MAE	Training R^2
1	200	0.17692	0.0313	0.14597	0.72169
2	400	0.13087	0.01713	0.09991	0.84772
3	600	0.11964	0.01431	0.09118	0.87273
4	800	0.11708	0.01371	0.08836	0.87811
5	1000	0.11048	0.0122	0.08152	0.89148
6	1200	0.11504	0.01323	0.08132	0.88234
7	1400	0.11596	0.01345	0.08452	0.88045
8	1600	0.10257	0.01052	0.07207	0.90647
9	1800	0.10224	0.01045	0.06954	0.90707
10	2000	0.10499	0.01102	0.08316	0.90199
10	2000	0.10224	0.01045	0.06954	0.90707

neural networks might not have considered all possible frameworks available, limiting the scope of the analysis.

Conclusion

Deep learning algorithms can be used to design systems that report data on patients and deliver warnings to medical applications or electronic health information if there are changes in the patient's health. These systems could be created using deep learning. This may help to verify that patients get the proper effective care at the proper time for each specific patient. A healthcare decision support system constructed on the Internet of Things and deep learning methods was presented. In the proposed system, we examined the capability of integrating deep learning technology into IoT features, such as self-diagnosis and real-time data transmission, which may help speed up. We have selected the suitable Neural Network structure number of fast-hidden layers and activation function classes to Design the e-health system. In addition, the e-health system relied on data from doctors to understand the Neural Network. In the -validation method, the total evaluation of the proposed healthcare system for diagnostics provides dependability under various patient conditions. Based on evaluation and simulation findings, the tanh function is more reliably stored in the neurons that use a dual hidden layer feed-forward NN. To overcome challenges, this study will integrate artificial intelligence with IoT. This study aims to determine the NN's optimal layer counts and activation function variations.

Future study

Future research could explore the impact of other factors on the neural network, such as the choice of optimization algorithm, regularization techniques, or hyperparameter tuning. The study could be extended to include a larger and more diverse dataset, considering different healthcare scenarios to validate the findings and improve the generalizability of the results. Further investigation could be conducted on the scalability and efficiency of the proposed techniques, considering larger datasets and more complex healthcare scenarios. Future studies could also focus on evaluating the robustness of the optimized neural network models to various types of noise or adversarial attacks in healthcare settings. Additionally, the research could investigate the interpretability of the optimized models and explore techniques to make the decision-making process more transparent and understandable to healthcare professionals. By addressing these limitations and conducting further research in these areas, it will be possible to enhance the understanding and applicability of the proposed techniques for speeding up neural network training while maintaining high accuracy in healthcare settings.

Contributorship: Conceptualization; SW, N.Z Jhanjhi; methodology; SW, BS, GX, supervision, N. Z Jhanjhi., SW;

software Analysis H. Dongyan and SW; validation, S. W., GX, RKM, and BS; formal analysis, SK, RKM, and NZ. Jhanjhi; investigation, H. Dongyan., GX, and BS; resources, RKM, BS, and H. Dongyan. Data curation, SWW; writing—original draft preparation, NZ. Jhanjhi; review and editing, SW, RKM, and NZ. Jhanjhi; visualization, SW and NZ. Jhanjhi; project administration, SW; funding acquisition, SW All authors have read and agreed to the published version of the manuscript.

Ethical statement: The dataset used in our research was obtained from Kaggle, and we understand the need to address ethical concerns even when using publicly available datasets. Here are some clarifications and additional information regarding the ethical aspects of our study.

Data Source and Anonymization: The patient data utilized in this study was collected and shared on Kaggle with due consideration for patient privacy and confidentiality. While Kaggle datasets often provide anonymized or de-identified data, we ensured that any potentially identifiable information, such as names or medical record numbers, was removed or hashed to protect patient identities.

Patient Consent and IRB Approval: Since our study utilized a publicly available dataset, it is important to note that the data was originally collected with the appropriate patient consent and Institutional Review Board (IRB) approvals, as per Kaggle's guidelines. We did not have direct access to patient information or the ability to seek additional consent.

Data Handling and Security: We followed best practices in data handling, storage, and security. The dataset was securely stored and accessed only by authorized team members involved in the research. We employed encryption and access controls to prevent unauthorized access.

Patient Privacy in Reporting: In the reporting of our research findings, we took great care to ensure that no personally identifiable information or patient-specific details were disclosed. Our focus was solely on the analysis and outcomes related to fraud in healthcare data prediction.

While we did not have direct involvement in obtaining patient consent or IRB approvals due to the nature of the dataset, we are committed to upholding ethical standards in healthcare research and recognize the importance of these considerations. We appreciate your feedback, and we will include a section in our manuscript to explicitly address these ethical aspects, reaffirming our commitment to responsible and ethical research practices.

Patients consent statement: This research is conducted using the online publicly available dataset, with the aim of finding the ideal number of best-hidden layers for the neural network as well as different activation function variations. The article also thoroughly analyzes how various frameworks can be used to create a comparison or fast neural networks. The final goal of the article is to investigate all such innovative techniques that allow us to speed up the training of neural networks without losing accuracy. This is completely technologically oriented research, which does not have any concern with the medical data or with patients' consent. This mainly focuses on how to speed up the training of models without losing accuracy.

Guarantor: Sobia Wassan is the guarantor.

ORCID iDs: Sobia Wassan https://orcid.org/0000-0001-5504-7496
N.Z. Jhanjhi https://orcid.org/0000-0001-8116-4733

References

1. Rössig A and Petkovic M. Advances in verification of ReLU neural networks. *J Glob Optim* 2021; 81: 109–152.
2. Bilic P, Christ P, Li HB, et al. The liver tumor segmentation benchmark (LITS). *Med Image Anal* 2023; 84: 102680.
3. Bolhasani H, Mohseni M and Rahmani A. Deep learning applications for IoT in health care: a systematic review. *Inform Med Unlocked* 2021; 23: 100550.
4. Malarvizhi Kumar P, Hong CS, Chandra Babu G, et al. Cloud-and IoT-based deep learning technique-incorporated secured health monitoring system for dead diseases. *Soft Comput* 2021; 25: 12159–12174.
5. Wassan S, Xi C, Jhanjhi N, et al. Effect of frost on plants, leaves, and forecast of frost events using convolutional neural networks. *Int J Distrib Sens Netw* 2021; 17: 15501477211053777.
6. Shaikh A, Faizullah S, Vasan D, et al. Slim-simple light-weight and intuitive multicast protocol for manets. *Int J Comput* 2014; 94: 45–47.
7. Wassan S, Suhail B, Mubeen R, et al. Gradient boosting for health IoT federated learning. *Sustainability* 2022; 14): 16842. ISSN.
8. Zikria YB, Afzal MK, Kim SW, et al. *Deep learning for intelligent IoT: Opportunities, challenges and solutions*. Elsevier, 2020, pp.50–53.
9. Wassan S, Chen X, Shen T, et al. Amazon Product sentiment analysis using machine learning techniques. *Rev Argentina de Clin Psicol* 2021; 30: 695.
10. Fahmy KAS, Yahya A, Zorkany MJJoE D, et al. A decision support healthcare system based on IoT and neural network technique. *J Eng Des Technol* 2022; 20: 727–748.
11. Ragab M, Algarni A, Bahaddad AA, et al. Machine learning applied to problem-solving in medical applications. *Comput Mater Contin* 2021; 69: 2277–2294.
12. Ullah N, Mugahed Al-Rahmi W, Alzahrani AI, et al. Blockchain technology adoption in smart learning environments. *Sustainability* 2021;13:1801.
13. Iranpak S, Shahbahrami A and Shakeri H. Remote patient monitoring and classifying using the internet of things

platform combined with cloud computing. *J Big Data* 2021; 8: 1–22.

14. Ganesh M, Pednekar P, Prabhuswamy P, et al. CNN-based android malware detection. In: 2017 international conference on software security and assurance (ICSSA) 2017, pp.60–65.

15. Vasan D, Alazab M, Venkatraman S, et al. MTHAEL: cross-architecture IoT malware detection based on neural network advanced ensemble learning. *IEEE Trans Comput* 2020; 69: 1654–1667.

16. Wassan S, Shen T, Xi C, et al. Customer experience towards the product during a coronavirus outbreak. *Behav Neurol* 2022; 2022: 4279346.

17. Vasan D, Alazab M, Wassan S, et al. IMCFN: image-based malware classification using fine-tuned convolutional neural network architecture. *Comput Netw* 2020; 10: 1016.

18. Humayun M, Jhanjhi NZ and Almotilag AJAS. Real-time security health and privacy monitoring for Saudi highways using cutting-edge technologies. *Appl Sci* 2022; 12: 2177.

19. Muzammal SM, Murugesan RK, Jhanjhi NZ, et al. SMTrust: proposing trust-based secure routing protocol for RPL attacks for IoT applications. In: 2020 International Conference on Computational Intelligence (ICCI) 2020; 305–310.

20. Namasudra S. An improved attribute-based encryption technique towards the data security in cloud computing. *Concurr Comput Pract Exp* 2019; 31: e4364.

21. Wassan S, Xi C, Jhanjhi N, et al. A smart comparative analysis for secure electronic websites. *Intell Autom Soft Comput* 2021; 29:187–199.

22. Dietz WH. Health consequences of obesity in youth: childhood predictors of adult disease. *Pediatrics* 1998; 101: 518–525.

23. Harerimana G, Kim JW, Yoo H, et al. Deep learning for electronic health records analytics. *IEEE Access* 2019; 7: 101245–101259.

24. Varshney M and Singh PJ. Optimizing nonlinear activation function for convolutional neural networks. *Signal Image Video Process* 2021; 15: 1323–1330.

25. Dörpinghaus J, Apke A, Lage-Rupprecht V, et al. *Data Exploration and validation on dense knowledge graphs for biomedical research*. Cornell University, 2019.

26. Kiliçarslan S and Celik M. RSigELU: a nonlinear activation function for deep neural networks. *Expert Syst Appl* 2021; 174: 114805.

27. Boullé N, Nakatsukasa Y and Townsend A. Rational neural networks. In: Part of Advances in Neural Information Processing Systems 33 (NeurIPS 2020) NeurIPS Proceedings, 2020, vol. 33, pp.14243–14253.

28. Zhang A, Lin Z, Wang B, et al. Nonlinear model predictive control of single-link flexible-joint robot using recurrent neural network and differential evolution optimization. *Electronics (Basel)* 2021; 10: 2426.

29. Alrawashdeh K and Purdy C. Fast activation function approach for deep learning based online anomaly intrusion detection. In: 2018 IEEE 4th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS) 2018, pp.5–13.

30. Zhao R, Yan R, Chen Z, et al. Deep learning and its applications to machine health monitoring. *Mech Syst Signal Process* 2019; 115: 213–237.

31. Nweke HF, Teh YW, Al-Garadi MA, et al. Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: state of the art and research challenges. *Expert Syst Appl* 2018; 105: 233–261.

32. Da Costa CA, Pasluosta CF, Eskofier B, et al. Internet of health things: toward intelligent vital signs monitoring in hospital wards. *Artif Intell Med* 2018; 89: 61–69.

33. Alam F, Mehmood R, Katib I, et al. Data fusion and IoT for smart ubiquitous environments: a survey. *IEEE Access* 2017; 5: 9533–9554.

34. Islam SR, Kwak D, Kabir MH, et al. The internet of things for health care: a comprehensive survey. *IEEE Access* 2015; 3: 678–708.

35. Lakshmanaprabu S, Mohanty SN, Krishnamoorthy S, et al. Online clinical decision support system using optimal deep neural networks. *Appl Soft Comput* 2019; 81: 105487.

36. Ho ESJS. Data security challenges in deep neural network for healthcare IoT systems. In: *Security and privacy preserving for IoT and 5G Networks*, Cham: Springer, 2022, pp.19–37.

37. Baek J-W and Chung KJIA. Context deep neural network model for predicting depression risk using multiple regression. *IEEE Access* 2020; 8: 18171–18181.

38. Swetha J. *Prediction of patient's health risk in critical care using a deep neural network*. Dublin Business School, 2020.

39. Khanna A, Selvaraj P, Gupta D, et al. Internet of things and deep learning enabled healthcare disease diagnosis using biomedical electrocardiogram signals. *Expert Syst* 2023; 40: e12864.