



Article

Artificial Neural Networks for the Prediction of Monkeypox Outbreak

Balakrishnama Manohar and Raja Das *

Department of Mathematics, School of Advanced Sciences, Vellore Institute of Technology (VIT),
Vellore 632014, India

* Correspondence: rajadasrkl@gmail.com

Abstract: While the world is still struggling to recover from the harm caused by the widespread COVID-19 pandemic, the monkeypox virus now poses a new threat of becoming a pandemic. Although it is not as dangerous or infectious as COVID-19, new cases of the disease are nevertheless being reported daily from many countries. In this study, we have used public datasets provided by the European Centre for Disease Prevention and Control for developing a prediction model for the spread of the monkeypox outbreak to and throughout the USA, Germany, the UK, France and Canada. We have used certain effective neural network models for this purpose. The novelty of this study is that a neural network model for a time series monkeypox dataset is developed and compared with LSTM and GRU models using an adaptive moment estimation (ADAM) optimizer. The Levenberg–Marquardt (LM) learning technique is used to develop and validate a single hidden layer artificial neural network (ANN) model. Different ANN model architectures with varying numbers of hidden layer neurons were trained, and the K-fold cross-validation early stopping validation approach was employed to identify the optimum structure with the best generalization potential. In the regression analysis, our ANN model gives a good R-value of almost 99%, the LSTM model gives almost 98% and the GRU model gives almost 98%. These three model fits demonstrated that there was a good agreement between the experimental data and the forecasted values. The results of our experiments show that the ANN model performs better than the other methods on the collected monkeypox dataset in all five countries. To the best of the authors' knowledge, this is the first report that has used ANN, LSTM and GRU to predict a monkeypox outbreak in all five countries.

Keywords: Hessian matrix; Levenberg–Marquardt model; K-fold cross-validation; regression analysis; machine learning; COVID-19; sigmoid function



Citation: Manohar, B.; Das, R.
Artificial Neural Networks for the
Prediction of Monkeypox Outbreak.
Trop. Med. Infect. Dis. **2022**, *7*, 424.
[https://doi.org/10.3390/
tropicalmed7120424](https://doi.org/10.3390/tropicalmed7120424)

Academic Editor: Matteo Riccò

Received: 9 October 2022

Accepted: 24 November 2022

Published: 8 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Almost all countries in the world were impacted by the COVID-19 pandemic that started in December 2019 in Wuhan, China. The onset of monkeypox in 2022, as reported by several nations, is another concern worldwide. The infectious condition known as monkeypox is brought on by the Zoonotic Ortho-poxvirus, a member of the Poxviridae family and the genus Ortho-poxvirus. It is closely linked to both cowpox and smallpox [1,2]. It is primarily transmitted by monkeys and rodents, although human-to-human transmission is also very common [3,4]. In a Copenhagen, Denmark lab in 1958, the virus was found for the first time in a monkey's body [5,6]. During a stepped-up drive to eliminate smallpox in 1970, the Democratic Republic of the Congo observed the first instance of monkeypox in a person [7,8]. Monkeypox often affects a large number of people who live close to tropical rainforests in central and western Africa. When a person comes into intimate touch with another infected individual, animal or object, the virus itself spreads. Direct bodily contact, animal bites, respiratory droplets or mucus from the eyes, nose or mouth can all spread the disease [9–11]. Fever, bodily pains and exhaustion are a few of the warning symptoms of a monkeypox infection in patients, with a red bump on the skin being the

long-term result [12,13]. Although COVID-19 has been found to be more contagious than monkeypox so far, the number of cases is still increasing. Only 50 cases of monkeypox were reported in West and Central Africa in 1990 [14,15]. However, in 2020, there were 5000 instances. In the past, monkeypox was thought to exist solely in Africa, but, in 2022, numerous additional non-African nations in Europe and the United States reported finding cases of the virus [16,17]. As a result, people are gradually becoming more and more anxious and afraid, which is frequently reflected in people's opinions on social media.

The first monkeypox outbreak was found in the UK on 6 May 2022. At the time of this writing, there are a total of 52,379 cases globally. The top five nations where it is more extreme are the USA (19,355 cases), Germany (3480 cases), the UK (3419 cases), France (3547) and Canada (1286) [18]. Despite the worldwide vaccination efforts, 17 people have died due to the virus, demonstrating that it is not fatal but is very infectious and evolves into other sub-variants. If not treated properly and controlled, the world may continue to see more new instances and deaths.

The Centers for Disease Control and Prevention (CDC) stated that there is currently no completely effective therapy for the monkeypox virus [19,20]. The CDC approved two oral medications, Brincidofovir and Tecovirimat, which had mostly been used to treat the smallpox virus but have now been utilized to treat the monkeypox virus in order to meet the urgent demand [21]. Although vaccinations against the monkeypox virus are available and have received FDA approval, they have not yet been used on humans in the United States. The vaccinations for the smallpox virus are used to treat the monkeypox virus in other nations [20].

Regression techniques using ANN methods are commonly used in order to predict future patient issues related to a particular disease [22]. Using the aforementioned methods, several research studies have been carried out to anticipate the range of devastation caused by diseases such as breast cancer, cardiovascular disease and COVID-19. The primary objective of this study is to forecast the confirmed cases of monkeypox in real-time. It also examines the early surveillance and prediction of the monkeypox pandemic [23]. Such real-time prediction systems could be very helpful for healthcare professionals and government agencies in guiding early responses for the very successful and timely management of these diseases. The decisions to control current possibilities can be made using these systems.

The actual number of monkeypox infection data indicates a set of observations that have been chronologically ordered. Time-series prediction techniques originated in statistics. Machine learning-based techniques, meta-predictors, structure-based techniques and ANNs also exist for this purpose [14,15]. ANNs are frequently employed for time series predictions [24]. One of the main benefits of an ANN approach is that it may be fed with unprocessed data that can automatically identify the necessary representation [25]. The ANN produces trustworthy outcomes based on a number of variables, including performance, accuracy, latency, speed, convergence and size. The USA, Germany, the UK, France and Canada are five of the nations with the highest number of monkeypox illnesses. No extensive works have been done on the monkeypox outbreak to the best of our knowledge.

The main contribution of this paper is that we have developed a forecasting model of the monkeypox time series dataset in the five countries mentioned above by employing state-of-the-art ANN models such as the LSTM and GRU. This study uses the ANN with an LM optimizer and the LSTM and GRU uses an ADAM optimizer. It further predicts the number of monkeypox cases directly resulting from this disease using NNs. These NNs used the existing datasets that contained all the available data related to the monkeypox epidemic in countries such as the USA, Germany, the UK, France and Canada.

The comparative study of these five countries will help the healthcare authorities to prepare for the necessary actions that need to be taken based on our model predictions. A summary of monkeypox from 6 May 2022 to 24 August 2022 in the USA, Germany, the UK, France and Canada has been considered.

The rest of the paper is organized as follows. The methods and materials are shown in Section 2, followed by a description of the ANN, LSTM and GRU structures, along with the

LM and ADAM optimizers. The results are presented and analyzed in Section 3. We come to certain conclusions in Section 4.

Related Works

Many researchers, including data scientists, have expended a lot of efforts to forecast the spread of this disease. By creating prediction models that emphasize the likely behaviors of this virus, data scientists may significantly advance the knowledge by improving our ability to forecast how the virus will spread. As a result, ANN models are thought of as precise tools that may aid in creating prediction models. In fact, several neural networks (NNs) have been developed in the past [26], as shown in Table 1.

Table 1. Summary of related work.

Study	Year	Technique	Input	Output	Results
A COVID-19 time series forecasting model based on MLP ANN [27]	2021	MLP and ANN	Daily confirmed cases	Next 20 days	More than 90%
Deep learning methods for forecasting COVID-19 time-series data: A comparative study [28]	2020	RNN, LSTM, Bi-LSTM and GRUs algorithms	Daily confirmed and recovered cases collected from six countries namely Italy, Spain, France, China, USA and Australia.	Forecasting of the number of new contaminated and recovered cases	VAE achieved MAPE values of 5.90%, 2.19%, 1.88%, 0.128%, 0.236% and 2.04%, respectively
Artificial Neural Networks for Prediction of COVID-19 in Saudi Arabia [24]	2021	ANN and MLPNN-PPA	Confirmed cases and deaths	The number of infected persons will increase in the coming days	The number of recoveries will be 2000 to 4000 per day.
Using Artificial Neural Network with Prey Predator Algorithm for Prediction of the COVID-19: the Case of Brazil and Mexico [29]	2021	ANN, PPA-BMLPNN and PPA-MMLPNN	Confirmed cases, recovered cases and deaths	The number of infected persons will increase in the coming days	The average active cases of COVID-19 in Brazil will go to 9×10^5 , with 1.5×10^5 recovered cases per day, and more than 6×10^5 as the total deaths.
Application of artificial neural networks to predict the COVID-19 outbreak [30]	2020	ANN-LM	Daily confirmed cases	The ANN-based model that takes into account the previous 14 days outperforms the other ones	The previous fourteen days for prediction are suggested to predict daily confirmed cases.
Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM [31]	2020	ARIMA, SVR, LSTM and Bi-LSTM	Daily confirmed cases and deaths	Prediction of confirmed cases and deaths	Bi-LSTM generates lowest MAE and RMSE values of 0.0070 and 0.0077, respectively
Time series prediction for the epidemic trends of COVID-19 using the improved LSTM deep learning method: Case studies in Russia, Peru and Iran [32]	2020	LSTM	Daily confirmed cases	Next 30 days	The proposed method can accurately analyze the trend of the epidemic.
Artificial neural networks for prediction of COVID-19 in India by using backpropagation [33]	2022	ANN-BP	Daily confirmed cases	The ANN-based model that takes into account the previous 14 days outperforms the other ones	The previous fourteen days for prediction are suggested to predict daily confirmed cases.
Monkeypox	Present	ANN-LM, LSTM-ADAM, GRU-ADAM	Daily confirmed cases	The number of infected persons will increase in the coming days	ANN-LM model (99%) perform better than LSTM and GRU (98%).

2. Methods and Materials

2.1. Artificial Neural Network

The ANN is a simple imitation of the neuron structure of the human brain [34]. Basic scalar messages, simple processing components, a high degree of interconnection and adoptive interaction between the units are the things which make them a type of multi-processor computer system [35]. Actually, the ANN provides a reasonably quick and flexible way of modelling, so it is appropriate for a rainfall-runoff prediction [36]. Layers of neurons make up an ANN. One or more hidden layers of neurons connect the input layer of neurons to the output layer of neurons. The interconnecting link between the neuron layers is made up of connection weights. This method changes its weights throughout the training phase to reduce the errors between the projected result and the actual output using the Back Propagation algorithm [37]. To get the best topology and weights, an ANN is trained using experimental data (called training data) and then evaluated with more experimental data (test-data). The accuracy of the model is checked using validation data. Bias refers to the weight that is provided directly to one neuron without being coupled to the prior neuron in specific circumstances. The most common type of ANN is the multilayer perceptron (MLP). It also has one or more hidden layers in the feed forward neural network.

A data structure that may operate like a neuron is constructed in ANNs with the use of programming skills. A node is a kind of data structure [38,39]. The network connecting these nodes is trained in this structure using a standard training technique like gradient descend. The nodes have two active states (on or off) and one inactive state (off or 0) in this memory or neural network, and each edge (synapse or link between nodes) has a weight. Negative weights inactivate or inhibit the following linked node (if active), whereas positive weights stimulate or activate the next inactive node [40–43]. The input d_p reaches the neuron c from the preceding neuron p in the ANN architecture. T_c is the total of the products of the inputs and their weights from Equation (1) [44], and w_{pc} is the weight of the input d_p with regard to cell c .

$$T_c = \sum w_{pc}d_p \tag{1}$$

The sigmoid function was chosen as the activation function and is applied to T_c . As a result, Equation (2) [44] is used to compute d_c :

$$d_c = \text{sigmoid}_c(T_c) \tag{2}$$

Likewise, the weights of the d_{cn} are w_{cn} , which is the output of c to n , which are computed. In a set, W is the sum of all the weights of the neural network, and $kw(x)$ is the neural network's output for input x and output y . The key purpose is to figure out these weights so that the error values between y and $kw(x)$ are reduced. That is, the aim is to reduce the cost function $E(W)$ (Equation (3)) to the smallest possible value [44]:

$$E(w) = \frac{1}{2} \sum_{i=1}^n (y_i - o_i)^2 \tag{3}$$

In this study, the ANN with LM optimizer [38], one of the most commonly used varieties of ANN, was utilized to forecast the epidemic. ANN was trained on a dataset using the LM algorithm. The optimum response was achieved by training the network with selected inner neurons. To minimize the cost function value, the results were calculated using the RMSE and correlation coefficient. The ANNs architecture is shown in Figure 1.

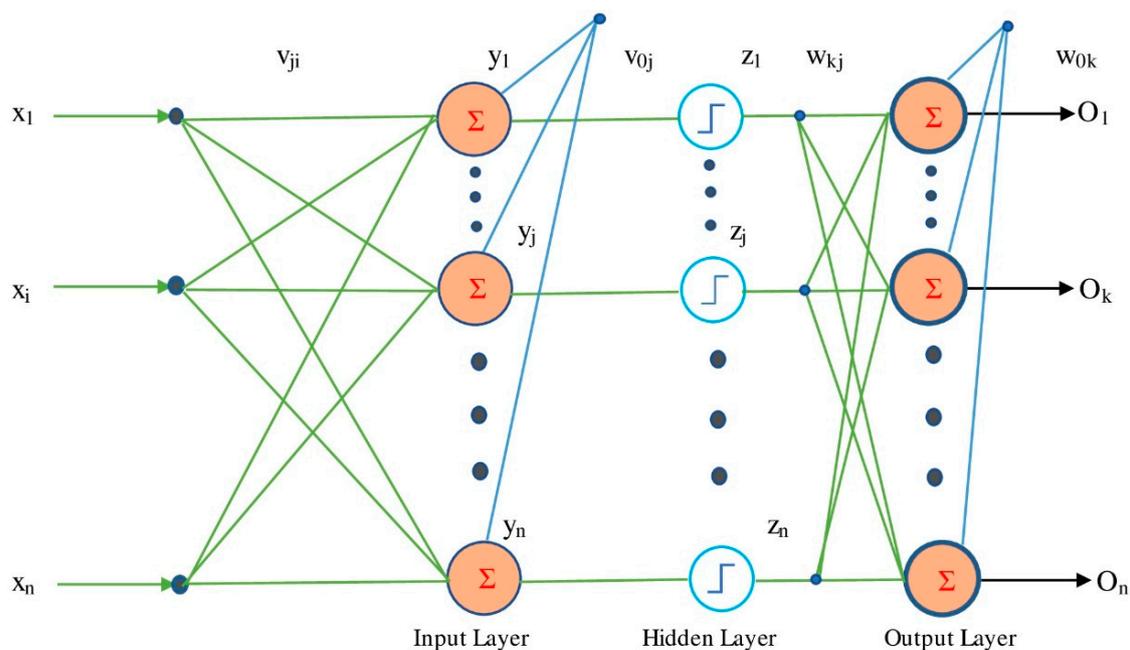


Figure 1. Architecture of an artificial neural network.

2.2. Levenberg–Marquardt Algorithm (LM)

The LM approach applies another approximation to the Hessian matrix in order to ensure that the estimated Hessian matrix $J^T J$ is invertible [45].

$$H \approx J^T J + \mu I \quad (4)$$

Here, μ is called the combination coefficient and is always positive and I is the identity matrix.

The components on the basic diagonal of the estimated Hessian matrix will be bigger than zero, as shown in Equation (5). As a result of this approximation (Equation (5)), the invertibility of matrix H can be guaranteed [46].

The update rule of the LM algorithm could be represented by merging Equations (4) and (5) as follows.

$$V_{k+1} = v_k - \left(J_k^T J_k + \mu I \right)^{-1} J_k e_k \quad (5)$$

Here, the weight vector is V and the error vector is e_k .

The LM algorithm shifts between the two techniques throughout the training phase, using a mix of the steepest descent algorithm and the Gauss–Newton algorithm. Equation (5) is the Gauss–Newton procedure, which is utilized when the combination coefficient is very tiny [47] (almost 0). Equation (5) approximates Equation (4), and the steepest descent approach is applied when the combination coefficient is extremely big. We explain the stepwise procedure of the LM algorithm in Figure 2.

Levenberg Marquardt (LM)

- Step 1. Start
- Step 2. Initialize the network weights (W) and biases (b) and other training parameters.
- Step 3. Start network training and calculate the error value for all input.
- Step 4. Calculate Jacobian Matrix JM (W).
- Step 5. Calculate ΔW by using eq.no. (5).
- Step 6. Calculate new error value by using weight updating formula:

$$\bar{W} = W + \Delta W$$
- Step 7. If error (Er) $<$ Er_{Min}
 - Step 7.1. Stop the network training and calculate output performance.
 - Otherwise
 - Step 7.2. Check if $Epoch \geq Epoch_{Min}$ if yes stop network training.
 - Otherwise
 - Go to step 5. (Update weights)
- Step 8. Increment epoch by 1:
 - $Epoch = Epoch + 1$
- Step 9. Go to Step 2 and continue the same procedure.
- Step 10. Stop.

Figure 2. Stepwise Levenberg–Marquardt algorithm procedure.

2.3. Long Short-Term Memory (LSTM)

The LSTM networks are made up of different gates that store information about the prior state. These data are either written, saved or retrieved from a cell that functions as a memory. When the cell reads, writes and erases using the gates that open and close, it determines whether to save the incoming information. They act depending on the signals they receive, blocking or passing on information according to its strength and import by filtering with their own weights. These weights are similar to those used to regulate

the input and hidden states via the network's training process [48]. This study proposes a LSTM network with an input layer, a hidden layer and an output layer. Figure 3 depicts an LSTM model with an input gate i_t , output gate o_t , forget gate f_t and cell state, c_t .

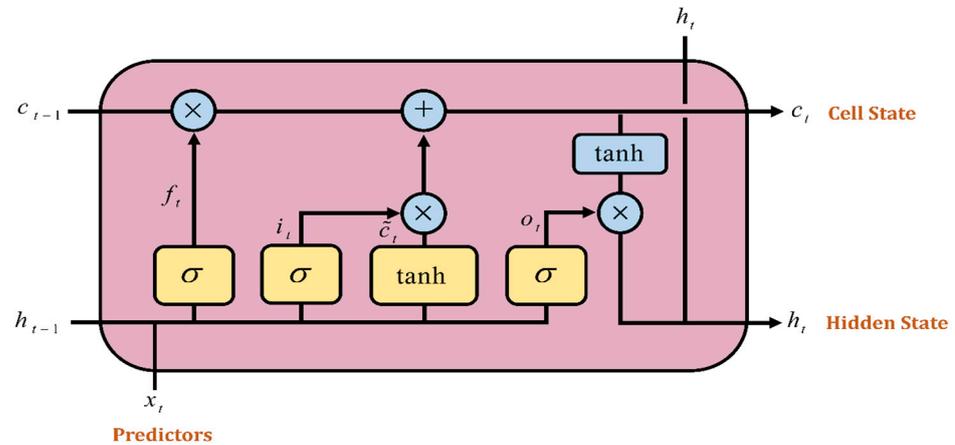


Figure 3. Long Short-Term Memory (LSTM) gates.

Memory blocks, which were designed to deal with disappearing gradients by remembering network parameters for long periods of time, are the most basic components of LSTM networks. Memory blocks in the LSTM architecture are analogous to digital systems' differential storage structures [49]. The activation functions (sigmoid and tansig) used by the gates in LSTM aids in the processing of information, and the output is either 0 or 1. Because we need to transfer only positive values to the following gates in order to produce a clear output, we employ sigmoid and tansig as activation functions [50]. The following Equations (6)–(11) represent the three gates of the LSTM network:

$$f_t = \sigma(x_t W_f + h_{t-1} u_f + b_f) \quad (6)$$

$$i_t = \sigma(x_t W_i + h_{t-1} u_i + b_i) \quad (7)$$

$$o_t = \sigma(x_t W_o + h_{t-1} u_o + b_o) \quad (8)$$

$$\bar{C}_t = \tanh(x_t W_C + h_{t-1} u_C + b_C) \quad (9)$$

$$C_t = \sigma(f_t + C_{t-1} + i_t + \bar{C}_t) \quad (10)$$

$$h_t = \tanh(C_t) \times o_t \quad (11)$$

Matrices W_q and u_q contain the weights of the input and recurrent connections, where the index can be the input gate i, output gate o, the forgetting gate f or the memory cell c, depending on the activation being calculated. C_t is not just a cell of an LSTM unit, but contains h cells of the LSTM units, while i_t , o_t and f_t represent the activations of, respectively, the input, output and forget gates, at time step t, where:

i_t = input gate function;

f_t = forget gate function;

o_t = output gate function;

h_t = hidden state function, also known as the output gate of the LSTM unit;

\bar{C}_t = cell input activation state;

C_t = cell state vector;

h_{t-1} = the result from the prior time step;

where W , u and b are the weight matrices and bias vector parameters which need to be learned during the training.

2.4. Gated Recurrent Unit (GRU)

The GRU is one of the variants of the RNN which was introduced by Cho et al. [28]. The update gate and resets gate are the two gates that the GRU utilizes. These gates employ activation functions similarly to the LSTM. The information from earlier time steps is added to the input data for time step t before being delivered to the update gate. This gate determines how much of this data flow needs to be sent on to the future and functions similarly to how the input gate and forget gate combine in the LSTM network. The reset gate determines how much of the previously computed state should be forgotten and stores the necessary data. The update gate determines what to collect from the previous steps and the current memory content to calculate the output of the current unit [49,51].

Figure 4 shows the internal architecture of a GRU unit cell. The mathematical Equations (12)–(15) are used to calculate these respective gates:

$$z_t = \sigma(x_t W^z + h_{t-1} u^z + b_z) \quad (12)$$

$$r_t = \sigma(x_t W^r + h_{t-1} u^r + b_r) \quad (13)$$

$$\bar{h}_t = \tanh(r_t \times h_{t-1} u + x_t W + b) \quad (14)$$

$$h_t = (1 - z_t) \times \bar{h}_t + z_t \times h_{t-1} \quad (15)$$

where W_z , W_r and W denote the weight matrices for the corresponding connected input vector. u_z , u_r and u represent the weight matrices of the previous time step and b_r , b_z and b are the bias. The σ denotes the logistic sigmoid function, r_t denotes the reset gate, z_t denotes the update gate and \bar{h}_t denotes the candidate hidden layer.

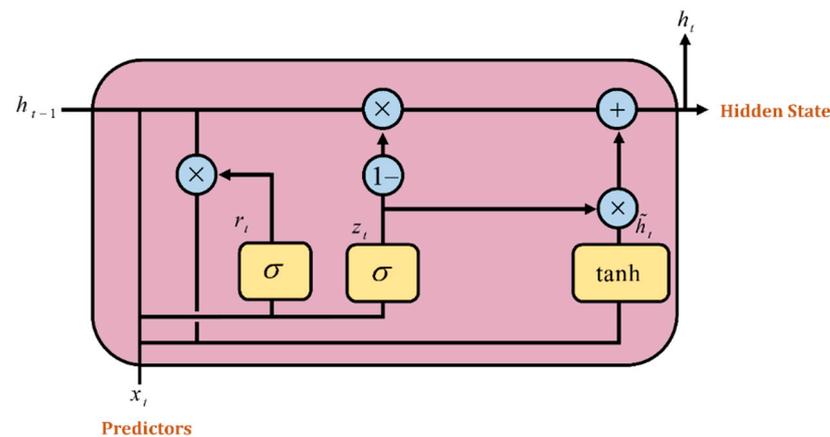


Figure 4. Gated Recurrent Unit (GRU) gates.

Deep learning networks are very sensitive to hyperparameters. The forecasted output will oscillate at high frequencies when the hyperparameters are wrongly configured [52]. The number of hidden neurons in the recurrent layers, the number of dropouts and the value of the learning rate are essential hyperparameters for GRU network models.

2.5. Adaptive Moment Estimation Optimization (ADAM)

Classification can be difficult when dealing with problems relating to the learning process. Several approaches have been proposed to help us arrive at an optimal learning level. The ADAM optimization algorithm is a deep learning extension of the stochastic gradient descent algorithm, which has recently been used in a variety of applications on the Internet of Things (IoT), text detection and so on [53].

ADAM is a famous optimizer that combines a gradient descent with momentum and the RMSprop optimizer [54]. The weights are updated using:

$$\theta_t = \theta_{t-1} - \left(\alpha / \sqrt{v'_t + \varepsilon} \right) \times m'_t \quad (16)$$

where v'_t and m'_t are the bias correction for the first and second moments, respectively:

$$m'_t = m_t / (1 - \beta_1^t) \quad (17)$$

$$v'_t = v_t / (1 - \beta_2^t) \quad (18)$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \times g_t \quad (19)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) \times g_t^2 \quad (20)$$

In Equation (19), m_t is the first moment that represents the running average of the gradients, whereas in Equation (20), v_t is the second moment that represents the running average of the squared gradients.

2.6. Network Modelling Process

The NN modelling procedure was carried out in two stages, including training and testing. The data must fall inside a narrow range to hasten the model convergence and improve the forecast accuracy. The study's input data were either in the tens of thousands or single digits. As a result, the min-max approach (Equation (21)), which requires that all input data points fall inside the range [0,1], was used using the following transformation.

$$X_{i,j} = \frac{X_i - X_{i,\min}}{X_{i,\max} - X_{i,\min}} \quad (21)$$

When " $x_{i,j}$ " refers to " x_i ", the actual value of the input variable ' i ' is normalized. The minimum and maximum values of the input variable ' i ' are, respectively, " $x_{i,\min}$ " and " $x_{i,\max}$ ". Similar to this, Equation (21) was used to normalize the target values so that they would fall inside the [0,1] operating range of the activation function.

The complete dataset was divided into two separate subsets after the data normalization. The workflow of the research methodology is shown in Figure 5.

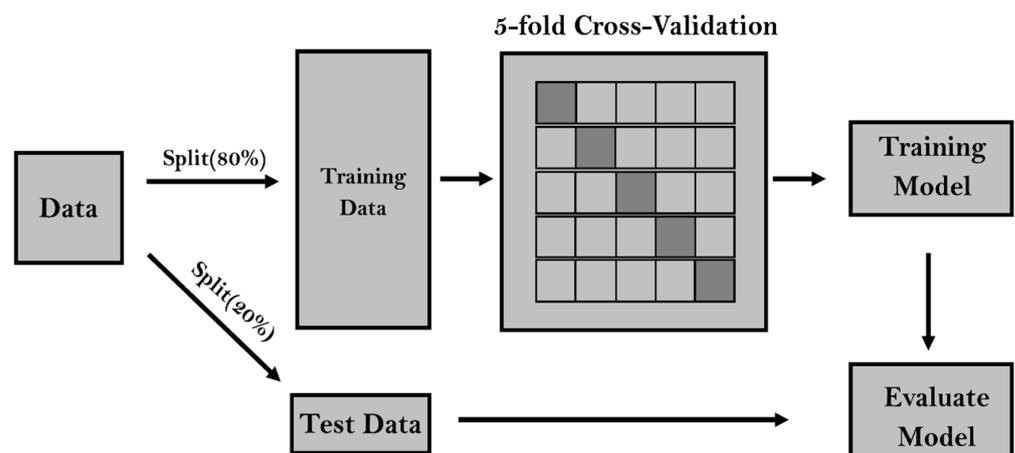


Figure 5. The flowchart of the research methodology workflow.

- (1) Training dataset: in order to reduce the error function, the model's synaptic weights were adjusted to correspond to the perfect number of hidden layer neurons. The cross-validation method was used to further split the training dataset into "K" subsets

in order to find the ideal number of iterations (or “epochs”) before the model training should be terminated.

- (2) Testing dataset: following the training phase, it was used to evaluate the model’s accuracy and forecasting capability.

2.7. Data Preparation

This study used daily confirmed cases data from the USA, Germany, the UK, France and Canada that are available from the “Global. Health” team website. First off, the researchers used data from 6 May 2022—the first case reported—to 31 August 2022 [55]. It was 2.5 MB in size and included 100 records, which is shown in Table 2. The following split of the data for that time period was utilized to discover the right parameters for the models: 80% for training and 20% for testing. Testing was the next stage after training. Figure 6 presents the trend of the daily confirmed cases of monkeypox for the five nations, which include the aforementioned countries.

Table 2. Data description.

Country	Data Description	Country-Code	WHO Region
United States	18 May 2022 to 24 August 2022	USA	Region of the Americas (AMR)
Germany	19 May 2022 to 24 August 2022	DE	European of Region (EUR)
United Kingdom	6 May 2022 to 24 August 2022	UK	European of Region (EUR)
France	19 May 2022 to 24 August 2022	FR	European of Region (EUR)
Canada	19 May 2022 to 24 August 2022	CA	Region of the Americas (AMR)

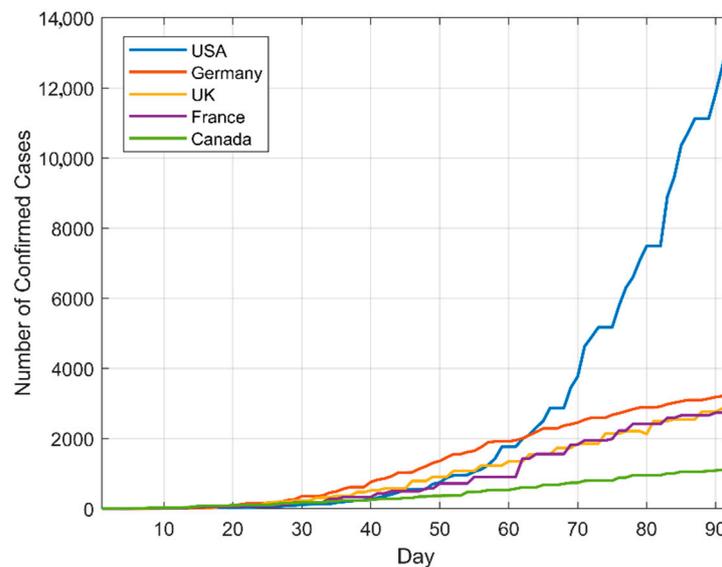


Figure 6. Daily confirmed cases for the USA, Germany, the UK, France and Canada—Monkeypox.

2.8. Network Model Evaluation

NN model training is an iterative procedure through which the model learns the input–output behavior. The LM learning method (Equations (4) and (5)) was utilized during the training stage. Two statistical indices were used to evaluate the performance of the model: the coefficient of determination (R^2), which is a measure of the model’s goodness-of-fit, and the root mean squared error (RMSE), which represents the square root of the average squared differences between the target value and the model output value. These two statistical indices are defined by Equations (22) and (23), respectively. The better the model fits the data, the R^2 is nearer to unity and the RMSE value is lower (nearer to zero). In other words, when R^2 equals 1.0 and RMSE equals 0, the model completely fits the data.

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{Y}_i - Y_i)^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2} \quad (22)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{Y}_i - Y_i)^2}{n}} \quad (23)$$

where \hat{Y} represents the predicted values and Y represents the actual values, and \bar{Y} represents the mean of the all the values and n denotes the number of values.

2.9. K-Fold Cross-Validation

The issue of when to end the training stage of an ANN model is a major conundrum since an overtrained model may perform poorly on an unknown dataset because it has learned the noise instead of the signals. One of the most popular techniques to prevent the model from overtraining is the stop-training criteria based on the k-fold cross-validation [56,57]. The k-fold cross-validation begins with the data being divided into K groups at random, after which the subsequent processes are carried out for each group.

- (1) Each fold in the “ K ” disjoint fold partition of the training dataset has the same number of samples.
- (2) In each of the “ K ” iterations, the model has trained on the first ($K-1$) folds.
- (3) The trained model is subsequently assessed on the final fold (also known as the validation fold) in order to calculate its RMSE.
- (4) The number of epochs versus the average RMSE is displayed on the validation folds.

The averaged RMSE typically falls during the early training phase and continues to rise after the network starts over-fitting the data. The RMSE should cease declining while the number of epochs rises, so that the training phase can be terminated.

2.10. Network Model Testing

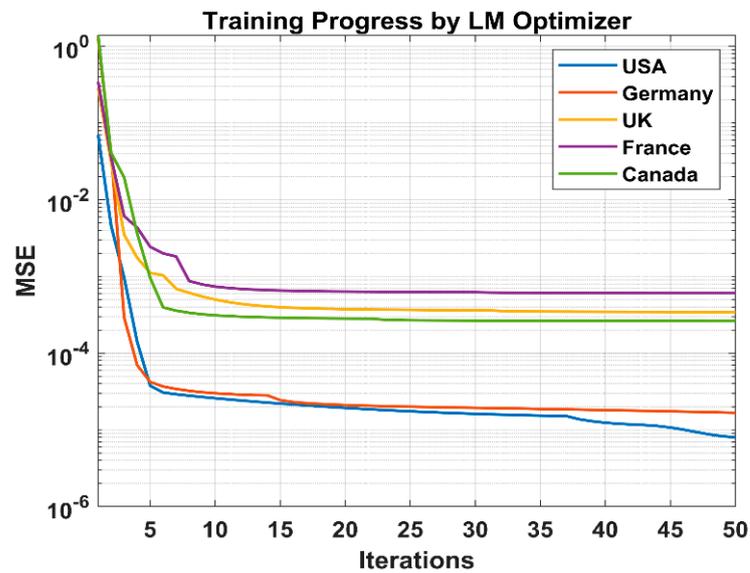
After the model training stage was finished, the trained model was evaluated against the test dataset (which was hidden throughout the training phase) to determine the model’s capacity for prediction. It should be noted that the output values are anti-normalized to their true values once the model training and testing phases are finished.

3. Results and Discussion

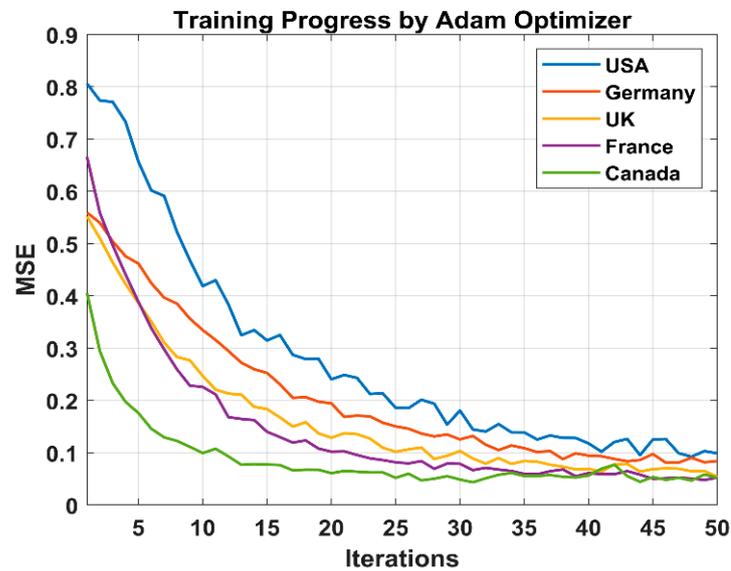
We analyzed the prediction performance of the three neural network models (ANN, LSTM and GRU) on data from five countries, namely the USA, Germany, the UK, France and Canada. The model performances are trained on data from 6 May to 9 August 2022 and evaluated using the test data from 10 August to 31 August 2021. The prediction performance of the models on the test data for all the models is shown in Figure 7.

Before beginning the hybrid modelling process, firstly a perceptron ANN with a single hidden layer and two hidden layers are developed [58,59]. For a sophisticated nonlinear issue, one or two hidden layers will be sufficient to train the ANN [60,61]. In addition, the Levenberg–Marquardt (LM) algorithm is used for the network training. The LM method has been shown to be one of the best and most flexible training algorithms, and as it avoids computing the Hessian Matrix, it could be viewed as the fastest backpropagation technique [15,58,62]. The standard approach described in the literature [63,64] is used to figure out the appropriate number of hidden neurons. In this regard, 1 to 24 ANN models (that means different hidden layers) are developed as shown in Table 3. Each model is categorized according to R^2 and the RMSE as a result of choosing the best option. A higher number is preferred for R^2 . As a result, under this ranking method, the model with the highest R^2 obtains the highest score (i.e., the maximum score is 24). On the other hand, a smaller RMSE number will be suitable. Therefore, the model with the lowest RMSE

value receives the highest ranking. Moreover, for each model, the overall rank is calculated by adding the two statistics for the training, validation and test stages, independently. Accordingly, in Table 3 the overall ranks attributable to the simulated models are calculated. As can be regarded, model no. 20 with 20 neurons has acquired the maximum total score. It can be claimed that the R^2 (RMSE) in this model reaches its maximum (minimum) in the training stage. From this point on, the R^2 will decrease as the number of neurons increases. Subsequently, based on the overall score, this model is picked as the optimal simulation.



(a)



(b)

Figure 7. Cont.

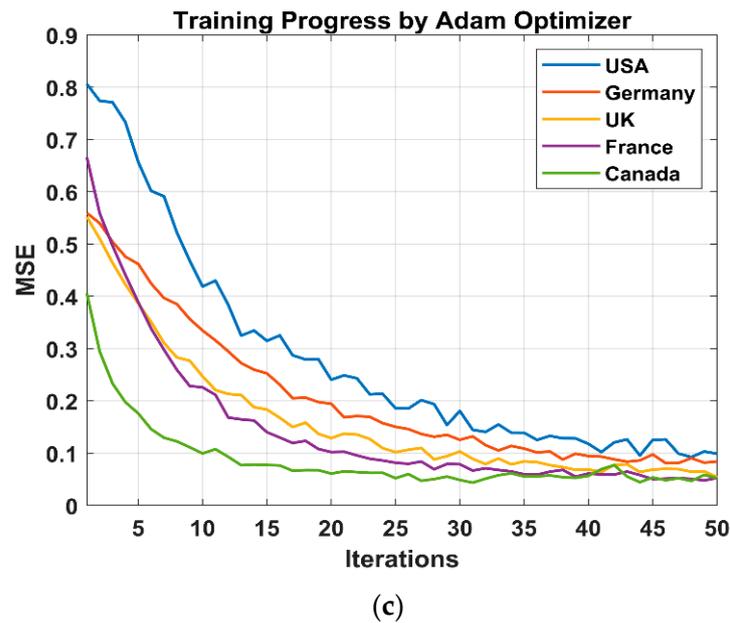


Figure 7. The performance plot of NN training by optimizer mean squared error (MSE) vs. iterations-monkeypox. (a) ANN; (b) LSTM; (c) GRU.

In Table 3, we have developed the ANN with a single hidden layer. Based on the highest score, the best architecture of the neural network has been decided for the USA dataset. As per Table 3, we have developed the ANN with two hidden layers (in Table 4) and the models are presented based on the top five highest score. As a result, in Tables 3 and 4, the best architecture of the neural network has been decided for the USA dataset. For Germany, the UK, France and Canada, we present the top five highest score models in Tables 5–8, respectively.

Tables 3–8 were compared, and we found that single hidden layers performed better than two hidden layers. Therefore, based on the ranking and overall score, the perfect ANN structure was determined for each country using ANN-LM models, i.e., for the USA 5-20-1 (5 neurons in the input layer, 20 neurons in the Hidden layer and 1 neuron in the output layer), for Germany (5-24-1), for the UK (5-18-1), for France (5-5-1) and for Canada (5-16-1).

After acquiring the ANN model's ideal structure for all the countries', i.e., 5-20-1 (the USA), 5-24-1 (Germany), 5-18-1 (the UK), 5-5-1 (France) and 5-16-1 (Canada) topology, it was then determined whether the model had been successfully trained or whether an undertraining or overtraining had taken place. A poor training performance is caused by undertraining, and the generalizability of the model might sometimes suffer from overtraining. In other words, the number of epochs at which the training phase is interrupted affects the model's performance and capacity to generalize. A five-fold cross-validation procedure was used on the training dataset to determine when it is optimal to cease the ANN model's training. The training MSE curve and the validation MSE curve, both calculated by the five-fold cross-validation, are shown in Figure 7a as functions of the number of training iterations.

Table 3. Selecting the optimal ANN model with respect to single hidden layer and neurons for the USA dataset.

Items	Neurons	No. of Hidden Layers	Train		Validation		Test		Train-Rank		Validation-Rank		Test-Rank		Overall Score
			R ²	RMSE	R ²	RMSE	R ²	RMSE	R ²	RMSE	R ²	RMSE	R ²	RMSE	
1	1	1	0.99944	0.00664	0.995686	0.019571	0.850461	0.12549	6	6	12	11	14	16	65
2	2	1	0.99959	0.00571	0.996186	0.018451	0.853831	0.124226	8	7	16	15	23	20	89
3	3	1	0.99875	0.00993	0.996379	0.017769	0.84821	0.125533	3	3	21	20	7	14	68
4	4	1	0.99976	0.00429	0.996808	0.016497	0.849201	0.123598	17	17	24	24	10	21	113
5	5	1	0.99958	0.00570	0.996302	0.018165	0.851358	0.125562	7	8	18	19	17	13	82
6	6	1	0.99966	0.00519	0.99552	0.019826	0.850677	0.124633	11	11	7	8	15	18	70
7	7	1	0.99843	0.01092	0.996327	0.01819	0.856681	0.123359	2	2	19	18	24	24	89
8	8	1	0.99964	0.00529	0.995946	0.018947	0.852853	0.124228	10	10	13	14	20	19	86
9	9	1	0.99960	0.00558	0.996128	0.01833	0.847895	0.125273	9	9	15	17	6	17	73
10	10	1	0.99979	0.00407	0.995653	0.019472	0.847847	0.125773	18	18	11	12	5	11	75
11	11	1	0.99987	0.00317	0.996481	0.01761	0.848805	0.125995	23	23	22	22	9	10	109
12	12	1	0.99935	0.00718	0.995547	0.019842	0.853296	0.12359	4	4	8	7	22	22	67
13	13	1	0.99981	0.00389	0.994944	0.021484	0.84724	0.129037	22	22	6	6	4	6	66
14	14	1	0.99979	0.00403	0.996191	0.018423	0.850882	0.125673	20	20	17	16	16	12	101
15	15	1	0.9993	0.00699	0.995982	0.019144	0.852618	0.126268	5	5	14	13	19	8	64
16	16	1	0.99979	0.00406	0.996796	0.016956	0.850439	0.126403	19	19	23	23	13	7	104
17	17	1	0.99971	0.00474	0.992316	0.027267	0.852996	0.129848	15	15	2	2	21	4	59
18	18	1	0.99972	0.00472	0.995592	0.019731	0.848797	0.126132	16	16	9	9	8	9	67
19	19	1	0.99970	0.00481	0.99563	0.019641	0.849848	0.125492	13	13	10	10	12	15	73
20	20	1	0.99980	0.00394	0.996354	0.01764	0.849419	0.123588	21	21	20	21	11	23	117
21	21	1	0.99987	0.00314	0.993076	0.024351	0.83232	0.131779	24	24	4	4	1	2	59
22	22	1	0.99971	0.00475	0.994395	0.022199	0.841461	0.129095	14	14	5	5	3	5	46
23	23	1	0.99828	0.01138	0.987112	0.035651	0.852588	0.13128	1	1	1	1	18	3	25
24	24	1	0.99967	0.00508	0.99283	0.025737	0.840422	0.133195	12	12	3	3	2	1	33

Table 4. Selecting the optimal ANN model with respect to two hidden layer and neurons for the USA dataset.

Items	Neurons	No. of Hidden Layers	Train		Validation		Test		Train-Rank		Validation-Rank		Test-Rank		Overall Score
			R ²	RMSE	R ²	RMSE	R ²	RMSE	R ²	RMSE	R ²	RMSE	R ²	RMSE	
1	5	2	0.99958	0.0057	0.996302	0.018165	0.851358	0.125562	7	8	18	19	17	13	82
2	11	2	0.99987	0.00317	0.996481	0.01761	0.848805	0.125995	23	23	22	22	9	10	109
3	12	2	0.99935	0.00718	0.995547	0.019842	0.853296	0.12359	4	4	8	7	22	22	67
4	16	2	0.99979	0.00406	0.996796	0.016956	0.850439	0.126403	19	19	23	23	13	7	104
5	19	2	0.9997	0.00481	0.99563	0.019641	0.849848	0.125492	13	13	10	10	12	15	73

Table 5. Selecting the optimal ANN model with respect to hidden layers and neurons for the Germany dataset.

Items	Neurons	No. of Hidden Layers	Train		Validation		Test		Train-Rank		Validation-Rank		Test-Rank		Overall Score
			R ²	RMSE	R ²	RMSE	R ²	RMSE	R ²	RMSE	R ²	RMSE	R ²	RMSE	
1	7	1	0.99981	0.00506	0.998549	0.012344	0.960484	0.065853	18	18	12	12	16	16	92
2	15	1	0.99972	0.00604	0.998841	0.011029	0.959775	0.066482	9	9	16	17	12	10	73
3	20	1	0.99977	0.00554	0.998526	0.012422	0.961158	0.065168	11	11	11	11	19	19	82
4	22	1	0.99983	0.00476	0.997861	0.014981	0.962513	0.064003	21	21	2	2	22	23	91
5	24	1	0.99993	0.00305	0.999307	0.008547	0.961561	0.06505	24	24	24	24	20	20	136
6	5	2	0.99971	0.00633	0.998842	0.011072	0.960826	0.065823	7	7	17	16	17	17	81
7	10	2	0.99982	0.00487	0.999036	0.010025	0.960202	0.065854	20	20	23	23	15	15	116
8	11	2	0.99979	0.00532	0.99891	0.010686	0.959587	0.066564	16	16	19	19	9	9	88
9	12	2	0.99988	0.00395	0.998736	0.011531	0.962649	0.06398	23	23	14	15	23	24	122
10	21	2	0.99980	0.00509	0.998238	0.013561	0.959688	0.066353	17	17	7	7	11	13	72

Table 6. Selecting the optimal ANN model with respect to hidden layers and neurons for the UK dataset.

Items	Neurons	No. of Hidden layers	Train		Validation		Test		Train-Rank		Validation-Rank		Test-Rank		Overall Score
			R ²	RMSE	R ²	RMSE	R ²	RMSE	R ²	RMSE	R ²	RMSE	R ²	RMSE	
1	4	1	0.99616	0.0174	0.997668	0.011402	0.631966	0.223179	12	12	15	15	13	15	82
2	7	1	0.99613	0.01742	0.998503	0.009161	0.632303	0.222859	10	11	22	22	14	16	95
3	15	1	0.9964	0.01678	0.998149	0.010213	0.634408	0.221676	18	19	20	20	18	20	115
4	18	1	0.99707	0.01515	0.998546	0.00899	0.631227	0.222802	22	23	24	24	12	17	122
5	20	1	0.9971	0.01516	0.996542	0.014373	0.635039	0.227021	23	22	6	4	19	2	76
6	6	2	0.99596	0.01784	0.997928	0.010815	0.629863	0.225646	8	8	18	18	8	4	64
7	9	2	0.99614	0.01742	0.998515	0.009107	0.630533	0.224038	11	10	23	23	11	10	88
8	14	2	0.99647	0.01658	0.997121	0.012745	0.637717	0.218903	20	20	14	13	22	24	113
9	16	2	0.99639	0.01693	0.997793	0.011191	0.635548	0.222772	17	17	16	16	20	18	104
10	22	2	0.99629	0.01723	0.997893	0.011138	0.638272	0.223483	16	13	17	17	24	13	100

Table 7. Selecting the optimal ANN model with respect to hidden layers and neurons for the France dataset.

Items	Neurons	No. of Hidden layers	Train		Validation		Test		Train-Rank		Validation-Rank		Test-Rank		Overall Score
			R ²	RMSE	R ²	RMSE	R ²	RMSE	R ²	RMSE	R ²	RMSE	R ²	RMSE	
1	3	1	0.99437	0.02372	0.997974	0.016372	0.949289	0.08408	10	8	20	20	20	21	99
2	5	1	0.99479	0.02259	0.998057	0.01597	0.948348	0.084555	15	15	22	22	18	17	109
3	6	1	0.99435	0.02356	0.997897	0.016645	0.946519	0.086316	9	10	16	16	15	14	80
4	14	1	0.99442	0.02325	0.99691	0.019969	0.947747	0.084289	11	11	9	10	16	19	76
5	21	1	0.9952	0.02176	0.993804	0.02864	0.955583	0.078299	18	19	2	2	23	23	87
6	7	2	0.99465	0.02308	0.997937	0.016588	0.949004	0.084624	13	14	17	18	19	16	97
7	9	2	0.99452	0.02319	0.998034	0.016004	0.944522	0.087533	12	12	21	21	12	11	89
8	16	2	0.99525	0.02161	0.996716	0.020902	0.945894	0.087204	19	20	8	7	14	13	81
9	18	2	0.99604	0.01973	0.997475	0.018082	0.942191	0.089217	23	24	14	14	7	8	90
10	20	2	0.996	0.01982	0.997155	0.019208	0.94329	0.08837	22	22	11	11	10	10	86

Table 8. Selecting the optimal ANN model with respect to hidden layers and neurons for the Canada dataset.

Items	Neurons	No. of Hidden Layers	Train		Validation		Test		Train-Rank		Validation-Rank		Test-Rank		Overall Score
			R ²	RMSE	R ²	RMSE	R ²	RMSE	R ²	RMSE	R ²	RMSE	R ²	RMSE	
1	2	1	0.99754	0.01453	0.999267	0.010368	0.923376	0.110758	9	9	18	18	23	23	100
2	6	1	0.99743	0.01477	0.99944	0.009105	0.920556	0.113625	4	4	24	24	12	9	77
3	10	1	0.99756	0.01446	0.999316	0.010048	0.923644	0.110944	10	11	20	20	24	22	107
4	16	1	0.99818	0.01248	0.999352	0.009834	0.921499	0.113232	19	19	23	23	19	13	116
5	24	1	0.99844	0.01156	0.998198	0.016301	0.92087	0.112905	23	23	2	2	16	16	82
6	7	2	0.9976	0.01435	0.99933	0.009967	0.920149	0.113964	12	13	22	22	9	8	86
7	8	2	0.99781	0.0137	0.99915	0.011158	0.920551	0.112932	14	14	14	14	11	15	82
8	20	2	0.99789	0.01341	0.99915	0.011127	0.920659	0.112564	17	17	15	15	14	18	96
9	21	2	0.9984	0.01172	0.998421	0.015152	0.922865	0.110642	22	22	3	3	21	24	95
10	23	2	0.99848	0.01228	0.999123	0.011263	0.922201	0.110967	24	20	13	13	20	21	111

3.1. Observing the Monkeypox Outbreak Using the ANN-LM Models in the Five Countries

The training performance of an ANN with an LM optimizer is shown in Figure 7a for all five countries. It is clear that at iteration four, the MSE drops drastically to its lowest level, after which the error essentially stays the same. With the best achievement being 0.00001 at iteration 14, the training continues until iteration 50. It is demonstrated that while the adjustment procedure is extremely slow, the LM optimizer always converges extremely quickly by observing all the training processes of an ANN with LM.

Regression plots are used to validate the network's performance, that show the network's output in terms of targets for training, validation, testing and overall datasets. The entire validation dataset is used by the ANN for training as well. Generally, in a regression plot, if the R-value is nearly 1, that means that the model is perfect (Figure 8a). As the R-value is 0.999 or above in each country, we can see that the fit is quite good (R-value) for the USA, Germany, the UK, France and Canada (0.99915, 0.99978, 0.99793, 0.99778 and 0.99917). Only the ANN-LM algorithm gives the best R-value of almost 0.99999 on the monkeypox outbreak.

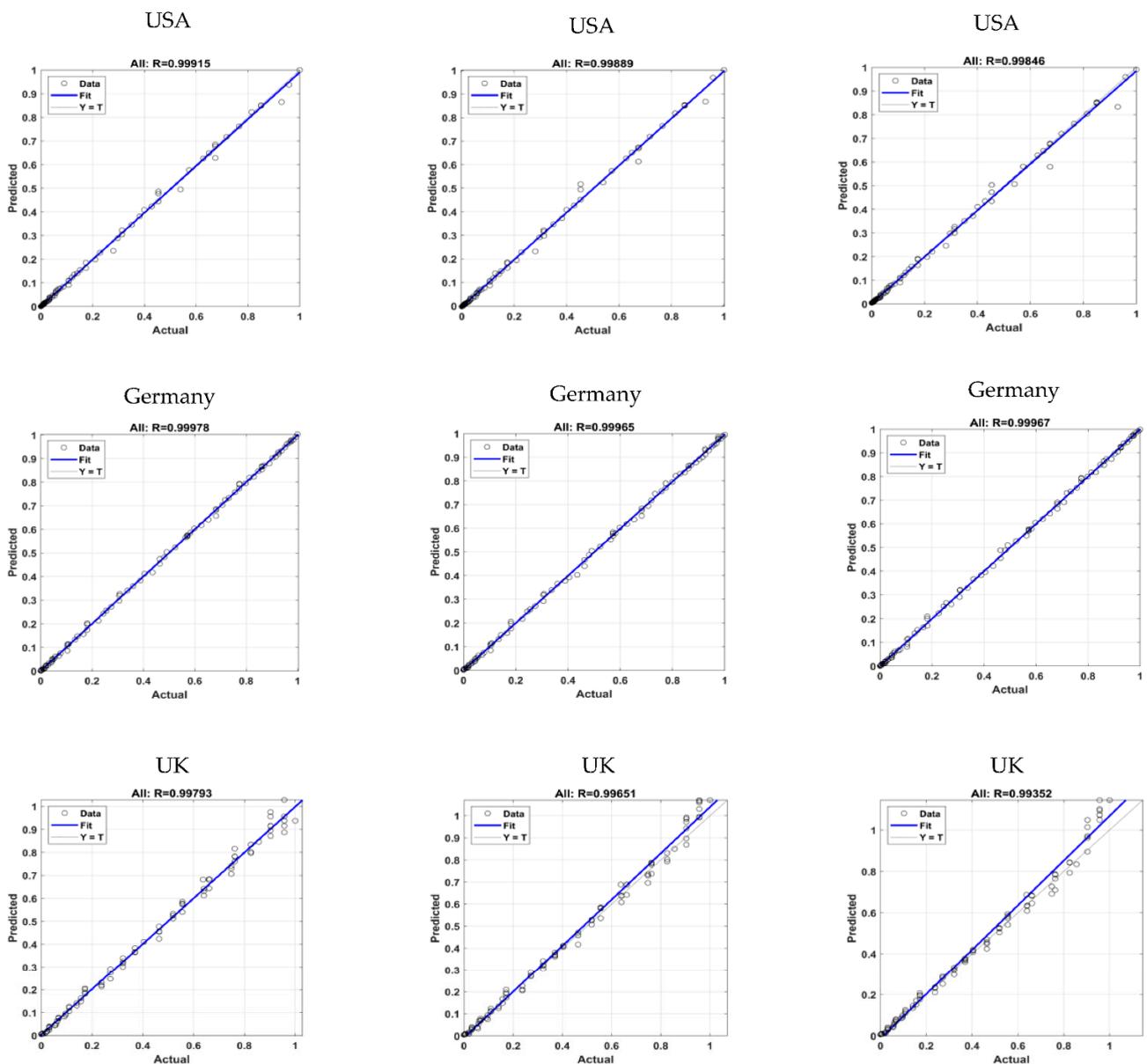


Figure 8. Cont.

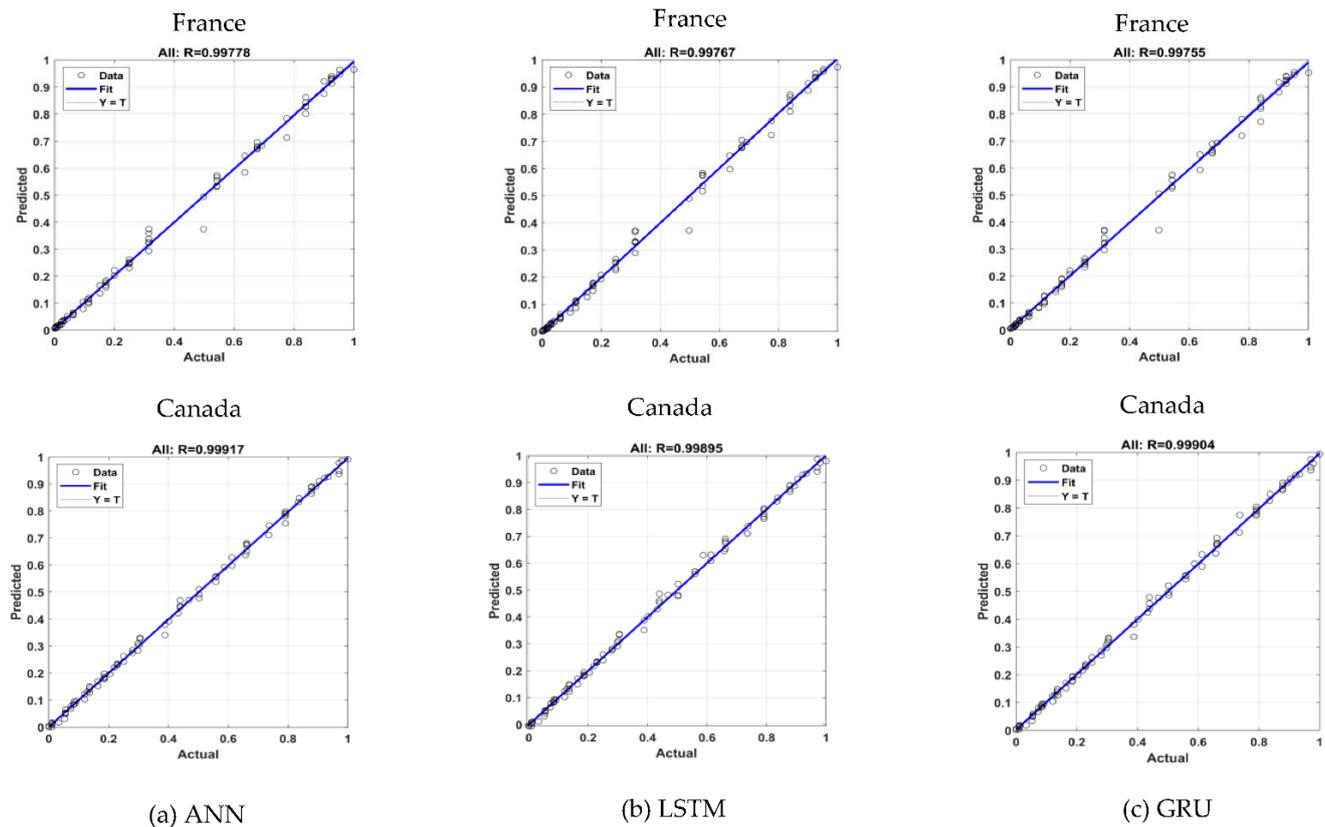


Figure 8. The proposed regression analysis (actual vs. predicted) of neural networks optimal models in predicting—monkeypox.

The error histogram is the histogram of the errors between the target values and predicted values after training a neural network. These can be negative as these error values indicate how predicted values differ from the target values. For the ANN-LM model, Figure 9a shows the training data as blue bars, testing data as red bars and validation data as green bars. The graphs are created using the error range (maximum negative error to maximum positive error), which is divided into 20 bins. The histogram makes it possible to spot outliers, which are data points where the fit is noticeably worse than the majority of the data. For the USA, Germany, the UK, France and Canada, Figure 9a shows that more errors in this instance are between -0.00111 and 0.00111 , -0.0012 and 0.0012 , -0.00383 and 0.00383 , -0.0006 and 0.0006 and -0.00112 and 0.00112 , respectively. However, there is one learning point (zero line) with 0.003851 , 0.001147 , 0.000899 , 0.008544 and 0.00274 errors, respectively. In this case, we can see that the ANN-LM method gives better results on a monkeypox outbreak.

In Figure 10a, the predicted and actual monkeypox incidence time trends are compared for the model's performance and accuracy. Plots that compared the observed (target) values to the model-calculated (output) values against time allowed us to observe how the network, outputs and targets responded to the inputs. Additionally, displayed are the errors that were discovered during the process. The ANN-LM model is capable of representing and simulating the desired output, and it provides a good representation of the overall trend of a monkeypox incidence. In addition, the majority of the estimation errors with respect to the time were between -0.02 and 0.02 for all five countries. The results indicate that our model selection was reasonably good.

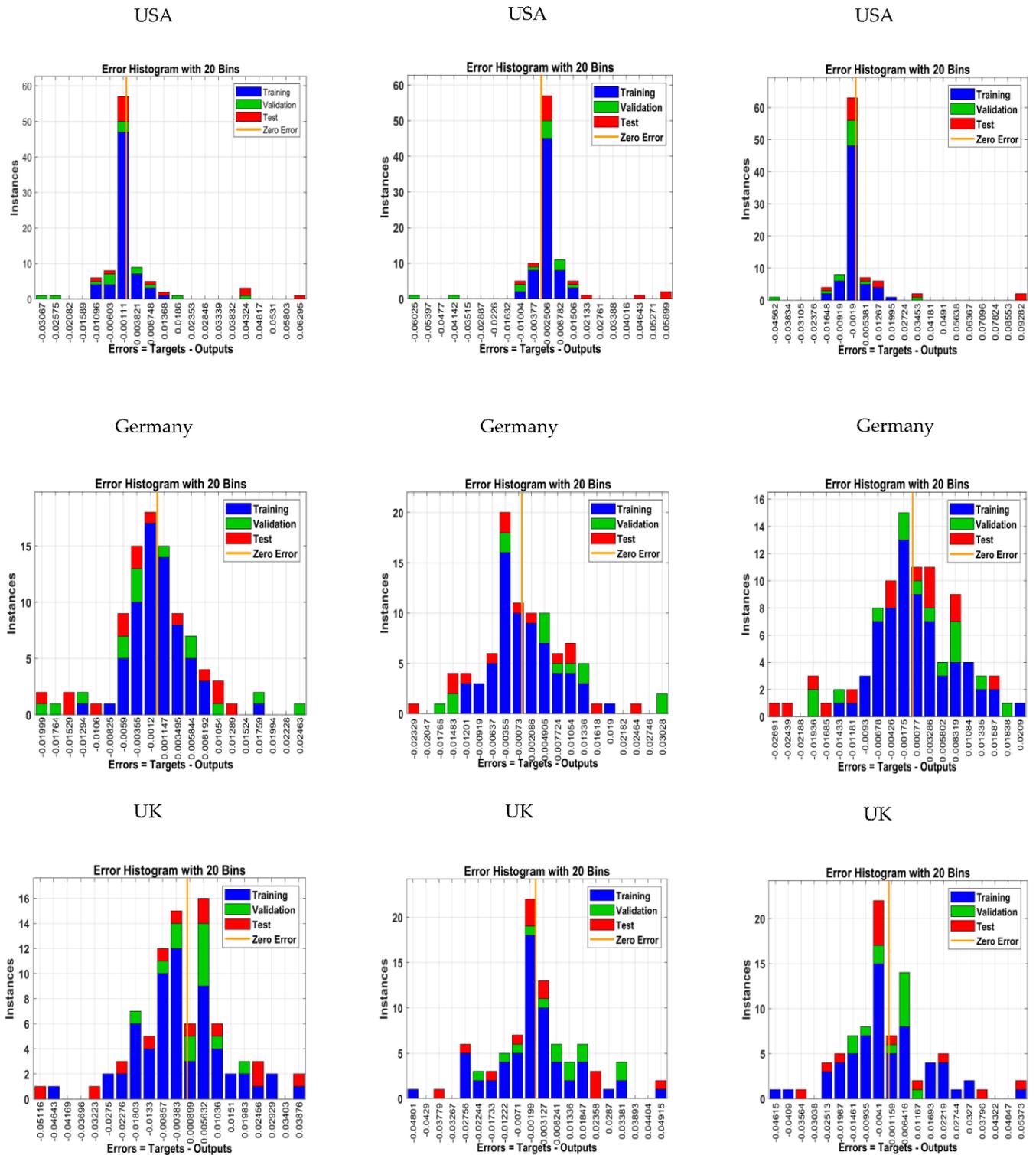


Figure 9. Cont.

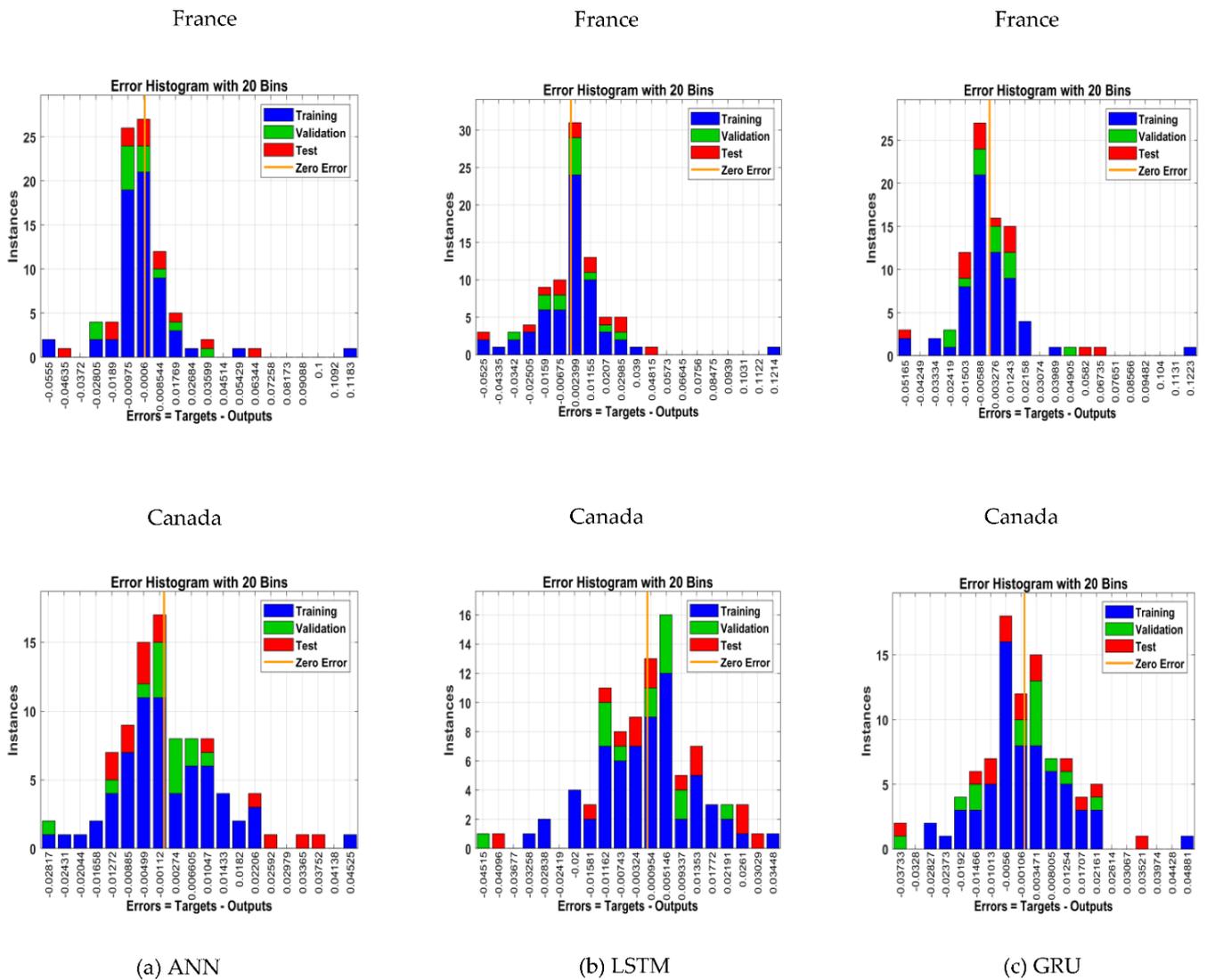


Figure 9. The error histogram of NN optimal model in predicting—monkeypox.

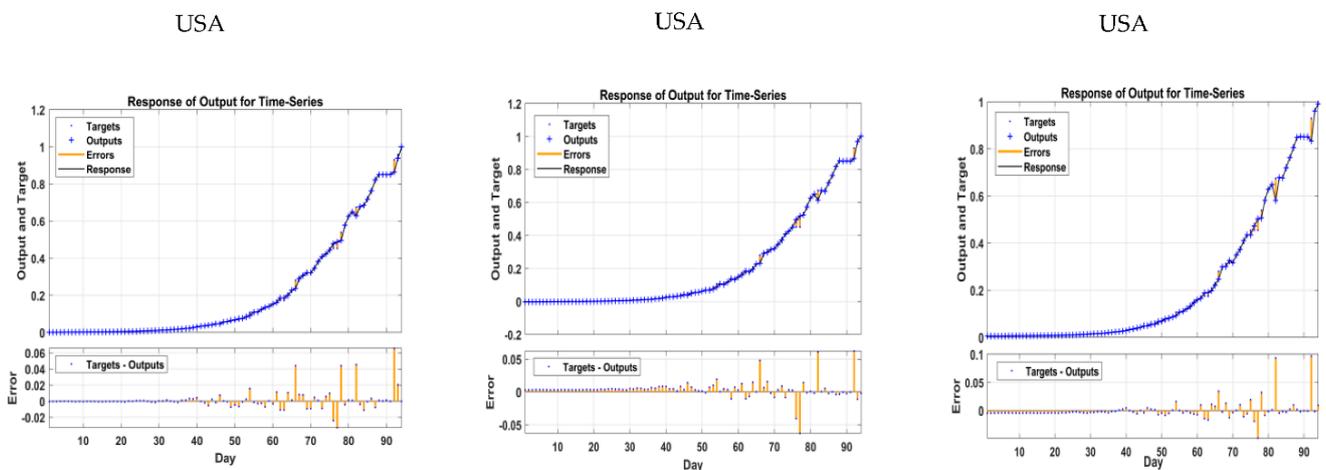


Figure 10. Cont.

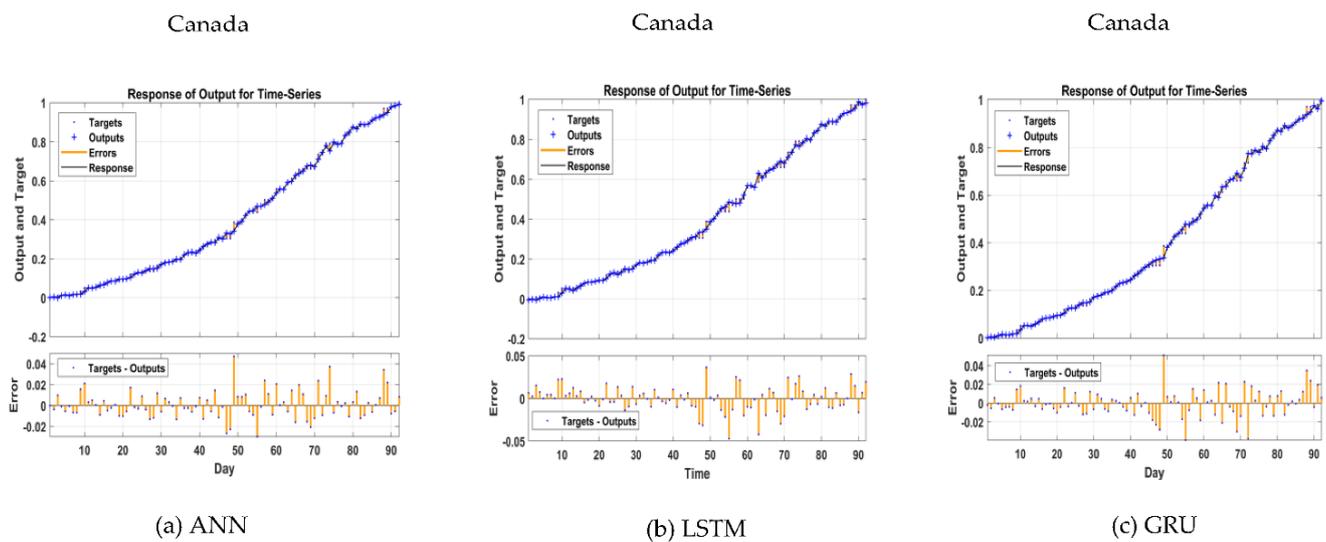


Figure 10. The responds plot of NN optimal model in predicting—Monkeypox.

3.2. Observing the Monkeypox Outbreak Using the LSTM-ADAM Models in the Five Countries

The training performance of an LSTM with an ADAM optimizer is shown in Figure 7b for all five countries in same figure. It is clear that at iteration 10, the MSE drops dramatically to its lowest level, after which the error essentially stays the same. With the best achievement being 0.1 at iteration 20, training continues until Iteration 50. It is demonstrated that while the adjustment procedure is extremely slow, the ADAM optimizer always converges extremely quickly by observing all the training processes of an LSTM with ADAM.

As the R-values are (0.99889, 0.99965, 0.99651, 0.99767 and 0.99895), we can see that the fit is quite good (Figure 8b) for all five countries. The LSTM-ADAM model gives the better R-value of almost 0.9888 on the monkeypox outbreak.

For the five countries, Figure 9b shows that more errors are between -0.00377 and 0.00377 , -0.00073 and 0.00073 , -0.00199 and 0.00199 , -0.00675 and 0.00675 and -0.00324 and 0.00324 , respectively. However, there is one learning point (zero line) with a 0.002506, 0.002086, 0.003127, 0.002399 and 0.000954 error, respectively. In this case, we can see that the LSTM-ADAM method provides better results on a monkeypox outbreak.

In Figure 10b, the predicted and actual monkeypox incidence time trends are compared for the model's performance and accuracy. The LSTM-ADAM model is capable of representing and simulating the desired output, and it provides a good representation of the overall trend of a monkeypox incidence. In addition, the majority of the estimation errors with respect to time were between -0.05 and 0.05 for all five countries. We conclude that our model selection was reasonable.

3.3. Observing the Monkeypox Outbreak Using the GRU-ADAM Models in the Five Countries

The training performance of GRU with an ADAM optimizer is shown in Figure 7c for all five countries. It is clear that at iteration 15, the MSE drops dramatically to its lowest level, after which the error essentially stays the same. With the best achievement being 0.1 at iteration 25, training continues until iteration 50. It is demonstrated that while the adjustment procedure is extremely slow, the ADAM optimizer always converges extremely quickly by observing all the training processes of GRU with ADAM.

As the R-values are (0.99846, 0.99967, 0.99352, 0.99755 and 0.99904), we can see that the fit is quite good (Figure 8c) for all five countries. The GRU-ADAM model gives the better R-value of almost 0.9888 on the monkeypox outbreak.

For the GRU-ADAM model, Figure 9c shows that more errors in this instance are between -0.0019 and 0.0019 , -0.00175 and 0.00175 , -0.0041 and 0.0041 , -0.00588 and 0.00588 and -0.00106 and 0.00106 , respectively. However, there is one learning point (zero

line) with a 0.005381, 0.00077, 0.001159, 0.003276 and 0.003471 error, respectively. In this case, we can see that the GRU-ADAM method provides better results on a monkeypox outbreak.

In Figure 10c, the predicted and actual monkeypox incidence time trends are compared for the model's performance and accuracy. The GRU-ADAM model is capable of representing and simulating the desired output, and it provides a good representation of the overall trend of a monkeypox incidence. In addition, the majority of the estimation errors with respect to the time were between -0.05 and 0.05 for all five countries. We conclude that our model selection was reasonable as a result.

4. Conclusions

The monkeypox epidemic has significantly impacted the lives of many people in several nations. This epidemic is becoming worse in certain places. There is currently no treatment for this infection, and there is little chance of accurately forecasting how severe it could be. So, in order to forecast this disease, we designed a neural network model using a time series monkeypox dataset and compared it with the LSTM and GRU models. We used the time series datasets, gathered from the five nations (the USA, Germany, the UK, France and Canada) impacted mostly by monkeypox. The LM learning technique was used to develop and validate a single hidden layer ANN model. Different ANN model architectures with varying numbers of hidden layer neurons were trained, and the K-fold cross-validation early stopping validation approach was employed to identify the optimum structure with the best generalization potential. In the regression analysis, the ANN-LM model gives a good R-value of almost 99%, the LSTM model gives almost 98% and the GRU model gives almost 98%. These three model fittings demonstrated that there was a good agreement between the experimental data and the forecasted values. The results of our experiments show that the ANN model performed better than the other methods on the collected monkeypox dataset in all five countries.

Author Contributions: Conceptualization, B.M. and R.D.; methodology, B.M.; validation, B.M. and R.D.; formal analysis, B.M.; investigation, R.D.; resources, B.M.; writing original draft preparation, B.M.; writing review and editing, B.M. and R.D.; visualization, R.D.; supervision, R.D.; project administration, R.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this paper is available in the references in Section 2.7.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. McCollum, A.M.; Damon, I.K. Human Monkeypox. *Clin. Infect. Dis.* **2014**, *58*, 260–267. [[CrossRef](#)] [[PubMed](#)]
2. Ola, P. What Predicts the Severity of Monkeypox and Other Infections? Available online: <https://osf.io/phgfq/> (accessed on 8 October 2022). [[CrossRef](#)]
3. Alakunle, E.; Moens, U.; Nchinda, G.; Viruses, M.O. Monkeypox Virus in Nigeria: Infection Biology, Epidemiology, and Evolution. *Viruses* **2020**, *12*, 1257. [[CrossRef](#)] [[PubMed](#)]
4. Rahmanian, V.; Jahanbin, K.; Jokar, M. Using twitter and web news mining to predict the monkeypox outbreak. *Asian Pac. J. Trop. Med.* **2022**, *15*, 236. [[CrossRef](#)]
5. Moore, M.J.; Rathish, B.; Zahra, F. *Monkeypox*; StatPearls: Treasure Island, FL, USA, 2022.
6. Chen, H.; Shen, J.; Wang, L.; Song, J. A Framework towards Data Analytics on Host–Pathogen Protein–Protein Interactions. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 4667–4679. [[CrossRef](#)]
7. Nolen, L.D.; Osadebe, L.; Katomba, J.; Likofata, J.; Mukadi, D.; Monroe, B.; Doty, J.; Hughes, C.M.; Kabamba, J.; Malekani, J.; et al. Extended Human-to-Human Transmission during a Monkeypox Outbreak in the Democratic Republic of the Congo. *Emerg. Infect. Dis.* **2016**, *22*, 1014–1021. [[CrossRef](#)]
8. Wang, L.; Shang, J.; Weng, S.; Aliyari, S.R.; Ji, C.; Cheng, G.; Wu, A. Genomic Annotation and Molecular Evolution of Monkeypox Virus Outbreak in 2022. *J. Med. Virol.* **2022**, *95*, e28036. [[CrossRef](#)]

9. Yinka-Ogunleye, A.; Aruna, O.; Ogoina, D.; Aworabhi, N.; Eteng, W.; Badaru, S.; Mohammed, A.; Agenyi, J.; Etebu, E.N.; Numere, T.-W.; et al. Reemergence of Human Monkeypox in Nigeria, 2017. *Emerg. Infect. Dis.* **2018**, *24*, 1149–1151. [[CrossRef](#)]
10. Nguyen, P.Y.; Ajisegiri, W.S.; Costantino, V.; Chughtai, A.A.; MacIntyre, C.R. Reemergence of Human Monkeypox and Declining Population Immunity in the Context of Urbanization, Nigeria, 2017–2020. *Emerg. Infect. Dis.* **2021**, *27*, 1007–1014. [[CrossRef](#)]
11. Thomassen, H.A.; Fuller, T.; Asefi-Najafabady, S.; Shiplacoff, J.A.G.; Mulembakani, P.M.; Blumberg, S.; Johnston, S.C.; Kisalu, N.K.; Kinkela, T.L.; Fair, J.N.; et al. Pathogen-Host Associations and Predicted Range Shifts of Human Monkeypox in Response to Climate Change in Central Africa. *PLoS ONE* **2013**, *8*, e66071. [[CrossRef](#)]
12. Thornhill, J.P.; Barkati, S.; Walmsley, S.; Rockstroh, J.; Antinori, A.; Harrison, L.B.; Palich, R.; Nori, A.; Reeves, I.; Habibi, M.S.; et al. Monkeypox Virus Infection in Humans across 16 Countries—April–June 2022. *N. Engl. J. Med.* **2022**, *387*, 679–691. [[CrossRef](#)]
13. Chen, H.; Zhao, H.; Shen, J.; Zhou, R.; Zhou, Q. Supervised Machine Learning Model for High Dimensional Gene Data in Colon Cancer Detection. In Proceedings of the 2015 IEEE International Congress on Big Data, New York, NY, USA, 27 June–2 July 2015; pp. 134–141.
14. Ahsan, M.M.; Uddin, M.R.; Farjana, M.; Sakib, A.N.; Momin, K.A.I.; Luna, S.A. Image Data Collection and Implementation of Deep Learning-Based Model in Detecting Monkeypox Disease Using Modified VGG16. *arXiv* **2022**, arXiv:2206.01862.
15. Saba, A.I.; Elsheikh, A.H. Forecasting the prevalence of COVID-19 outbreak in Egypt using nonlinear autoregressive artificial neural networks. *Process. Saf. Environ. Prot.* **2020**, *141*, 1–8. [[CrossRef](#)]
16. Chavda, V.P.; Vora, L.K.; Apostolopoulos, V. Monkeypox: A new face of outbreak. *Expert Rev. Vaccines* **2022**, *21*, 1537–1540. [[CrossRef](#)]
17. Mohbey, K.K.; Meena, G.; Kumar, S.; Lokesh, K. A CNN-LSTM-Based Hybrid Deep Learning Approach to Detect Sentiment Polarities on Monkeypox Tweets. *arXiv* **2022**, arXiv:2208.12019.
18. Mathieu, E.; Spooner, F.; Dattani, S.; Ritchie, H.; Roser, M. Monkeypox. Our World in Data. 2022. Available online: <https://ourworldindata.org/monkeypox> (accessed on 8 October 2022).
19. Petersen, B.W.; Harms, T.J.; Reynolds, M.G.; Harrison, L.H. Use of Vaccinia Virus Smallpox Vaccine in Laboratory and Health Care Personnel at Risk for Occupational Exposure to Orthopoxviruses—Recommendations of the Advisory Committee on Immunization Practices (ACIP), 2015. *MMWR Morb. Mortal. Wkly. Rep.* **2016**, *65*, 257–262. [[CrossRef](#)] [[PubMed](#)]
20. Ogoina, D.; Iroezindu, M.; James, H.I.; Oladokun, R.; Yinka-Ogunleye, A.; Wakama, P.; Otike-Odibi, B.; Usman, L.M.; Obazee, E.; Aruna, O.; et al. Clinical Course and Outcome of Human Monkeypox in Nigeria. *Clin. Infect. Dis.* **2020**, *71*, e210–e214. [[CrossRef](#)]
21. Adler, H.; Gould, S.; Hine, P.; Snell, L.B.; Wong, W.; Houlihan, C.F.; Osborne, J.C.; Rampling, T.; Beadsworth, M.B.; Duncan, C.J.; et al. Clinical features and management of human monkeypox: A retrospective observational study in the UK. *Lancet Infect. Dis.* **2022**, *22*, 1153–1162. [[CrossRef](#)] [[PubMed](#)]
22. Meng, F.; Uversky, V.N.; Kurgan, L. Comprehensive Review of Methods for Prediction of Intrinsic Disorder and Its Molecular Functions. *Cell. Mol. Life Sci.* **2017**, *74*, 3069–3090. [[CrossRef](#)]
23. Al-Qaness, M.A.A.; Ewees, A.A.; Fan, H.; El Aziz, M.A. Optimization Method for Forecasting Confirmed Cases of COVID-19 in China. *J. Clin. Med.* **2020**, *9*, 674. [[CrossRef](#)]
24. Hamadneh, N.N.; Khan, W.A.; Ashraf, W.; Atawneh, S.H.; Khan, I.; Hamadneh, B.N. Artificial Neural Networks for Prediction of Covid-19 in Saudi Arabia. *Comput. Mater. Contin.* **2021**, *66*, 2787–2796. [[CrossRef](#)]
25. Wang, L.; Wang, Z.; Qu, H.; Liu, S. Optimal Forecast Combination Based on Neural Networks for Time Series Forecasting. *Appl. Soft Comput.* **2018**, *66*, 1–17. [[CrossRef](#)]
26. Apaydin, H.; Feizi, H.; Sattari, M.T.; Colak, M.S.; Shamshirband, S.; Chau, K.-W. Comparative Analysis of Recurrent Neural Network Architectures for Reservoir Inflow Forecasting. *Water* **2020**, *12*, 1500. [[CrossRef](#)]
27. Borghi, P.H.; Zakordonets, O.; Teixeira, J.P. A COVID-19 time series forecasting model based on MLP ANN. *Procedia Comput. Sci.* **2021**, *181*, 940–947. [[CrossRef](#)] [[PubMed](#)]
28. Zeroual, A.; Harrou, F.; Dairi, A.; Sun, Y. Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study. *Chaos Solitons Fractals* **2020**, *140*, 110121. [[CrossRef](#)] [[PubMed](#)]
29. Hamadneh, N.N.; Tahir, M.; Khan, W.A. Using Artificial Neural Network with Prey Predator Algorithm for Prediction of the COVID-19: The Case of Brazil and Mexico. *Mathematics* **2021**, *9*, 180. [[CrossRef](#)]
30. Niazkar, H.R.; Niazkar, M. Application of artificial neural networks to predict the COVID-19 outbreak. *Glob. Health Res. Policy* **2020**, *5*, 50. [[CrossRef](#)]
31. Kim, M.H.; Kim, J.H.; Lee, K.; Gim, G.Y. The Prediction of COVID-19 Using LSTM Algorithms. *Int. J. Netw. Distrib. Comput.* **2021**, *9*, 59–74. [[CrossRef](#)]
32. Wang, P.; Zheng, X.; Ai, G.; Liu, D.; Zhu, B. Time Series Prediction for the Epidemic Trends of COVID-19 Using the Improved LSTM Deep Learning Method: Case Studies in Russia, Peru and Iran. *Chaos Solitons Fractals* **2020**, *140*, 11021. [[CrossRef](#)]
33. Manohar, B.; Das, R. Artificial Neural Networks for Prediction of COVID-19 in India by Using Backpropagation. *Expert Syst.* **2022**, e13105. Available online: <https://onlinelibrary.wiley.com/doi/full/10.1111/exsy.13105> (accessed on 14 June 2022). [[CrossRef](#)]
34. Geirhos, R.; Janssen DH, J.; Schütt, H.H.; Rauber, J.; Bethge, M.; Wichmann, F.A. Comparing Deep Neural Networks against Humans: Object Recognition When the Signal Gets Weaker. *arXiv* **2017**, arXiv:1706.06969.
35. Rashidi, M.; Ali, M.; Freidoonimehr, N.; Nazari, F. Parametric Analysis and Optimization of Entropy Generation in Unsteady MHD Flow over a Stretching Rotating Disk Using Artificial Neural Network and Particle Swarm optimization algorithm. *Energy* **2013**, *55*, 497–510. [[CrossRef](#)]

36. Asadi, S.; Shahrabi, J.; Abbaszadeh, P.; Tabanmehr, S. A new hybrid artificial neural networks for rainfall–runoff process modeling. *Neurocomputing* **2013**, *121*, 470–480. [CrossRef]
37. Aichouri, I.; Hani, A.; Bougherira, N.; Djabri, L.; Chaffai, H.; Lallahem, S. River Flow Model Using Artificial Neural Networks. *Energy Procedia* **2015**, *74*, 1007–1014. [CrossRef]
38. Mirzazadeh, A.; Abdollahpour, S.; Mahmoudi, A.; Bukat, A. Intelligent Modeling of Material Separation in Combine Harvester’s Thresher by ANN. *Int. J. Agric. Crop Sci.* **2012**, *4*, 1767–1777.
39. Khalesi, S.; Mahmoudi, A.; Hosainpour, A.; Alipour, A. Detection of Walnut Varieties Using Impact Acoustics and Artificial Neural Networks (ANNs). *Mod. Appl. Sci.* **2011**, *6*, 43. [CrossRef]
40. Hossain, M.A.; Ayodele, B.V.; Cheng, C.K.; Khan, M.R. Artificial neural network modeling of hydrogen-rich syngas production from methane dry reforming over novel Ni/CaFe₂O₄ catalysts. *Int. J. Hydrogen Energy* **2016**, *41*, 11119–11130. [CrossRef]
41. Taghavifar, H.; Mardani, A. Wavelet Neural Network Applied for Prognostication of Contact Pressure between Soil and Driving Wheel. *Inf. Process. Agric.* **2014**, *1*, 51–56. [CrossRef]
42. Sharabiani, V.R.; Kassar, F.H.; Gilandeh, Y.A.; Ardabili, S.F. Application of Soft Computing Methods and Spectral Reflectance Data for Wheat Growth Monitoring. *Iraqi J. Agric. Sci.* **2019**, *50*, 1064–1076.
43. Reshadsedghi, A.; Mahmoudi, A. Detection of Almond Varieties Using Impact Acoustics and Artificial Neural Networks. *Int. J. Agric. Crop Sci.* **2013**, *6*, 1008–1017.
44. Hassoun, M.H. *Fundamentals of Artificial Neural Networks*; The MIT Press: Cambridge, MA, USA, 1995; Volume 84.
45. Padhi, P.C.; Mahapatra, S.S.; Yadav, S.N.; Tripathy, D.K. Performance Characteristic Prediction of WEDM Process Using Response Surface Methodology and Artificial Neural Network. *Int. J. Ind. Syst. Eng.* **2014**, *18*, 433–453. [CrossRef]
46. Chalisgaonkar, R.; Kumar, J.; Pant, P. Prediction of Machining Characteristics of Finish Cut WEDM Process for Pure Titanium Using Feed Forward Back Propagation Neural Network. *Mater. Today Proc.* **2020**, *25*, 592–601. [CrossRef]
47. Janmanee, P.; Kumjng, S. A Study of Tungsten Carbide Surfaces during the Electrical Discharge Machining Using Artificial Neural Network Model. *Int. J. Appl. Eng. Res.* **2017**, *12*, 3214–3227.
48. Chimmula, V.K.R.; Zhang, L. Time series forecasting of COVID-19 transmission in Canada using LSTM networks. *Chaos Solitons Fractals* **2020**, *135*, 109864. [CrossRef] [PubMed]
49. Shahid, F.; Zameer, A.; Muneeb, M. Predictions for COVID-19 with Deep Learning Models of LSTM, GRU and Bi-LSTM. *Chaos Solitons Fractals* **2020**, *140*, 110212. [CrossRef]
50. Liu, C.-H.; Gu, J.-C.; Yang, M.-T. A Simplified LSTM Neural Networks for One Day-Ahead Solar Power Forecasting. *IEEE Access* **2021**, *9*, 17174–17195. [CrossRef]
51. Yu, S.; Han, R.; Zheng, Y.; Gong, C. An Integrated AMPSO-CLSTM Model for Photovoltaic Power Generation Prediction. *Front. Energy Res.* **2022**, *10*, 264. [CrossRef]
52. Al-Haija, A.; Gui, G.; Su, R.; Yu, R.; Alsulami, A.A.; Abu Al-Haija, Q.; Alqahtani, A.; Alsini, R. Symmetrical Simulation Scheme for Anomaly Detection in Autonomous Vehicles Based on LSTM Model. *Symmetry* **2022**, *14*, 1450.
53. Amoudi, G.; Albalawi, R.; Baothman, F.; Jamal, A.; Alghamdi, H.; Alhothali, A. Arabic rumor detection: A comparative study. *Alex. Eng. J.* **2022**, *61*, 12511–12523. [CrossRef]
54. GitHub—Globaldothealth/Monkeypox: Monkeypox 2022 Repository. Available online: <https://github.com/globaldothealth/monkeypox> (accessed on 8 October 2022).
55. Fedotenkova, M. Extraction of Multivariate Components in Brain Signals Obtained during General Anesthesia. Ph.D. Thesis, Université de Lorraine, Metz, France, 2016.
56. Salehi, R.; Lestari, R.A.S. Predicting the performance of a desulfurizing bio-filter using an artificial neural network (ANN) model. *Environ. Eng. Res.* **2020**, *26*, 200462. [CrossRef]
57. Shaibani, M.J.; Emamgholipour, S.; Moazeni, S.S. Investigation of Robustness of Hybrid Artificial Neural Network with Artificial Bee Colony and Firefly Algorithm in Predicting COVID-19 New Cases: Case Study of Iran. *Stoch. Environ. Res. Risk Assess. Res. J.* **2022**, *36*, 2461–2476. [CrossRef]
58. Silitonga, P.; Bustamam, A.; Muradi, H.; Mangunwardoyo, W.; Dewi, B.E. Comparison of Dengue Predictive Models Developed Using Artificial Neural Network and Discriminant Analysis with Small Dataset. *Appl. Sci.* **2021**, *11*, 943. [CrossRef]
59. Karsoliya, S. Approximating Number of Hidden Layer Neurons in Multiple Hidden Layer BPNN Architecture. *Int. J. Eng. Trends Technol.* **2012**, *3*, 714–717.
60. Laureano-Rosario, A.E.; Duncan, A.P.; Mendez-Lazaro, P.A.; Garcia-Rejon, J.E.; Gomez-Carro, S.; Farfan-Ale, J.; Savic, D.A.; Muller-Karger, F.E. Application of Artificial Neural Networks for Dengue Fever Outbreak Predictions in the Northwest Coast of Yucatan, Mexico and San Juan, Puerto Rico. *Trop. Med. Infect. Dis.* **2018**, *3*, 5. [CrossRef] [PubMed]
61. Gülcü, Ş. Training of the feed forward artificial neural networks using dragonfly algorithm. *Appl. Soft Comput.* **2022**, *124*, 109023. [CrossRef]
62. Yarsky, P. Using a genetic algorithm to fit parameters of a COVID-19 SEIR model for US states. *Math. Comput. Simul.* **2021**, *185*, 687–695. [CrossRef]
63. Zorlu, K.; Gokceoglu, C.; Ocakoglu, F.; Nefeslioglu, H.; Acikalin, S. Prediction of uniaxial compressive strength of sandstones using petrography-based models. *Eng. Geol.* **2008**, *96*, 141–158. [CrossRef]
64. Koopialipour, M.; Noorbakhsh, A.; Ghaleini, E.N.; Armaghani, D.J.; Yagiz, S. A new approach for estimation of rock brittleness based on non-destructive tests. *Nondestruct. Test. Evaluation* **2019**, *34*, 354–375. [CrossRef]