

Hyperparameters analysis of long short-term memory architecture for crop classification

Madiha Sher¹, Nasru Minallah^{1,2}, Tufail Ahmad^{2,3}, Waleed Khan^{1,2}

¹Department of Computer Systems Engineering, University of Engineering and Technology Peshawar, Peshawar, Pakistan

²National Center for Big Data and Cloud Computing (NCBC), University of Engineering and Technology Peshawar, Peshawar, Pakistan

³Department of Computer Sciences, University of Engineering and Technology Peshawar, Peshawar, Pakistan

Article Info

Article history:

Received Oct 7, 2022

Revised Dec 19, 2022

Accepted Dec 21, 2022

Keywords:

Crop classification

Grid search

Hyperparameters tuning

Multispectral

Remote sensing

ABSTRACT

Deep learning (DL) has seen a massive rise in popularity for remote sensing (RS) based applications over the past few years. However, the performance of DL algorithms is dependent on the optimization of various hyperparameters since the hyperparameters have a huge impact on the performance of deep neural networks. The impact of hyperparameters on the accuracy and reliability of DL models is a significant area for investigation. In this study, the grid Search algorithm is used for hyperparameters optimization of long short-term memory (LSTM) network for the RS-based classification. The hyperparameters considered for this study are, optimizer, activation function, batch size, and the number of LSTM layers. In this study, over 1,000 hyperparameter sets are evaluated and the result of all the sets are analyzed to see the effects of various combinations of hyperparameters as well as the individual parameter effect on the performance of the LSTM model. The performance of the LSTM model is evaluated using the performance metric of minimum loss and average loss and it was found that classification can be highly affected by the choice of optimizer; however, other parameters such as the number of LSTM layers have less influence.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Nasru Minallah

Department of Computer Systems Engineering, University of Engineering and Technology Peshawar

Peshawar, Khyber Pakhtunkhwa, Pakistan

National Center for Big Data and Cloud Computing (NCBC), University of Engineering and Technology

Peshawar, Pakistan

Email: n.minallah@uetpeshawar.edu.pk

1. INTRODUCTION

Remote sensing (RS) data and crop classification techniques provide information useful for crop yield estimation and prediction. Latest satellite constellations can acquire satellite image time series (SITS) data with high levels of accuracy in terms of spectral, spatial, and temporal characteristics. SITS contains multi-scale data about land cover information and the phenological cycles of the crops, considered important for distinguishing between different crop types [1]. During the last several years, deep learning (DL) has been the most efficient approach for crop classification and has outperformed traditional machine learning techniques. Many DL architectures are successfully used for crop identification using RS data such as recurrent neural networks (RNN) [2], long-short-term-memory (LSTM) [3], and convolutional neural networks (CNN) [4].

LSTM is a type of RNN that retains information for a long time due to its recurrent back-propagation. The recurrent structure allows for retrieving complex, nonlinear relationships and the gating system regulates the flow of data in and out of the LSTM cell. Hence, RNN efficiently deals with sequential time series data [5]. In contrast to traditional neural network-based models, there are feedback connections in LSTM allowing input

sequences processing of any length and are widely used in time-series data classification, processing, and prediction. LSTM has been demonstrated to be effective for multiple agricultural tasks, including crop classification, crop monitoring, and crop area estimation [6]. However, it is often cumbersome to attain precise results using LSTM networks or any other DL architecture. The DL model is like a black box with numerous hyperparameters, including batch size, number of network layers, and activation function type, that can be adjusted accordingly. The performance of a DL model is largely dependent on the tuning of its hyperparameters, which can make a tremendous difference between ordinary and exceptional results.

The hyperparameter selection process is considered critical and must be found before training any model. Nonetheless, there is no set rule to rank hyperparameters based on their impact on performance. Hyperparameter tuning demands practice and in some cases, intelligent search. The two most commonly used methods for hyperparameter selection in DL are; manual searching and using searching algorithms. Researchers often employ ad-hoc manual tuning to pick the hyperparameter sets for testing performance. The manual method uses designs of multiple architectures with different hyperparameters and goes through an iterative process to create a high-performance region of the hyperparameter space. This method is popular among researchers because it is simple and quickly arrives at a reasonable solution once suitable hyperparameters are realized. However, this strategy hinders the reproduction of the same quality results on new data, especially for non-experts. The inefficiency of manual hyperparameter optimization justifies the need for its formalization. Reproducing the quality results on new data can be achieved via search algorithms. So far, several different search algorithms have been devised for hyperparameter tuning, such as grid search, random search [7], and Bayesian search [8]. Grid search performs an extensive search over the manually defined subset of possible hyperparameters. Even though other search algorithms may have better features and may take lesser time, grid search remains the most viable approach due to its simplicity and accurate outcomes [9]. Other hyperparameter optimization techniques, such as random search, and Bayesian search can considerably cut the search time. However, these methods either do not guarantee the best results or are unable to parallelize, unlike grid search [9].

Literature study shows that only a few hyperparameters matter for most of the datasets. In this study, 4 different hyperparameters are evaluated and the choices per hyperparameter are limited to fairly small, manually selected sets. This study evaluates almost one thousand combinations of hyperparameters. This study's key achievement is the determination of impactful hyperparameters and the analysis of their impact on the overall model performance based on the grid search hyperparameter optimization results. The remainder of the paper is organized as follows. Section 2 gives an overview of the region considered for this study and how the data was put together for inputting to the DL model. Next, we describe the gating mechanism of LSTM and explain the architecture used for this study. We then present an overview of how the grid search algorithm explores the whole search space and explain the hyperparameters considered for this study. In the next two sections, an analysis of grid search algorithm results is done and we finally conclude the best hyperparameters set for the LSTM model when used for RS data.

2. MATERIALS AND METHODS

2.1. Study area

The study region, Charsadda, is located in Khyber Pakhtunkhwa province of Pakistan and it expands from 71.28 E to 71.59 E and from 34.00 N to 34.30 N in Figure 1. This location has vast areas of arable land and a good variety of vegetation. The season that is selected for this study is Autumn. Major crops planted in this area are maize and sugarcane. Other than these, reed, maize, grasslands, trees, and yam plants were found in the study area. Ground survey data was collected from the study area in September 2020.

2.2. Remotely sensed data

Five dates were chosen for satellite imagery in this investigation to capture the reflectance of crops at various growth stages. Considering the phenological cycle of the sugarcane and maize, the dates chosen were the 15th, 20th, 23rd, and 30th of September, 2020, and the 5th of October, 2020. The availability of cloud-free imagery was also a factor in choosing these dates. The 23rd of September's imagery is of Planet-Scope, while the rest of the imagery is from sentinel-2. The timeline of selected imagery of the study area is shown in Figure 2.

2.2.1. Planet scope

The planet-scope system holds the record for being the largest commercial satellite array in existence with 120 satellites in the orbit, gathering daily images of the whole continent of planet earth. These satellites can capture imagery with a spatial resolution of 3 meters having four multispectral bands: blue, green, red, and near-infrared [10], considered sufficient for analyzing and tracking variations in the growth of plants. This

study is based on three bands of planet-scope namely red, green, and near-infrared. Planet-scope imagery of the pilot region for this study was collected on the 23rd of September 2020.

2.2.2. Sentinel 2

The Copernicus Sentinel-2 mission is made up of two satellites that are positioned at opposite ends of the same orbit in the same sun-synchronous orbit [11]. Its vast sweep width (290 kilometers), high spectral resolution (10 to 60 meters), and high revisit frequency aid in observing the Earth’s surface changes. With a focus on crop classification, three spectral bands namely red, green, and near-infrared (that were the same as planet-scope) for four sentinel-2 images were considered.

2.3. Data preparation

Ground truth data from the study area was collected at the end of September 2020 using an locally designed geo-survey mobile application [12]. The amount of data collected during the ground survey of all the classes considered for this study is shown in Table 1. The satellite imagery for this study was acquired according to the timeline mentioned in Figure 2. After acquiring the satellite imagery, the first step in data preparation was resampling. All the sentinel-2 bands used for this study have 10 meters resolutions which were resampled to 3 meters resolutions using bilinear interpolation to make them stackable with planet-scope imagery. Normalized difference vegetation index (NDVI), a popular index that experts use in RS [13], is calculated and stacked with individual images. Next, all the images were stacked together and the resulting data was standardized for further use by the DL model.

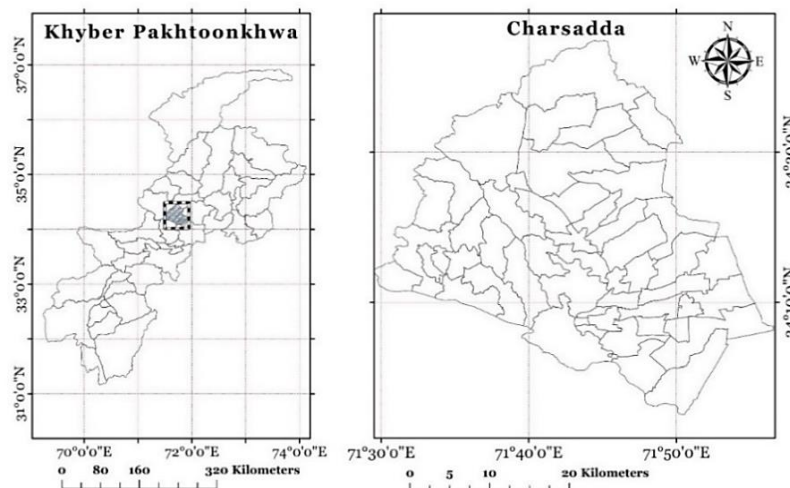


Figure 1. Locality map of the study area

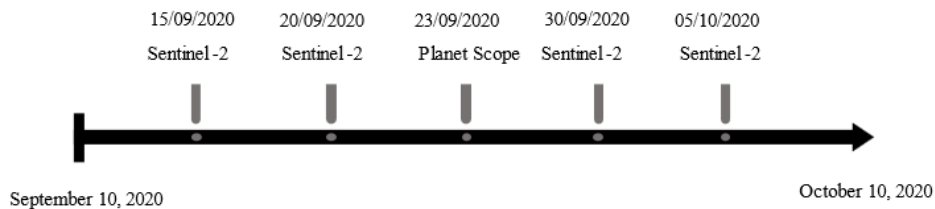


Figure 2. Timeline of the acquired imagery of the regions of interest

| Table 1. Number of pixels for each class | | |
|--|-------------------|------------------|
| Label | Class | Number of Pixels |
| 0 | Urban/Barren Land | 32124 |
| 1 | Trees | 30094 |
| 2 | Other Vegetation | 12280 |
| 3 | Sugarcane | 144523 |
| 4 | Maize | 42191 |
| 5 | Water | 43218 |

2.4. Methods

2.4.1. LSTM deep neural network model

The LSTM architecture is a type of RNN that was developed to solve error back-flow problems [5]. The LSTM is a chain-structured model as illustrated in Figure 3, where the crucial cell state represented by the horizontal line at the top of the figure has minimal linear interactions as it passes down the entire chain. The data flow through this line without being altered. The LSTM controls the cell state (C_t) and outputs (h_t) with three gates: a “forget” gate, an “input” gate, and an “output” gate. These gates control the amount of data that is allowed to pass through and the amount of data that is set aside.

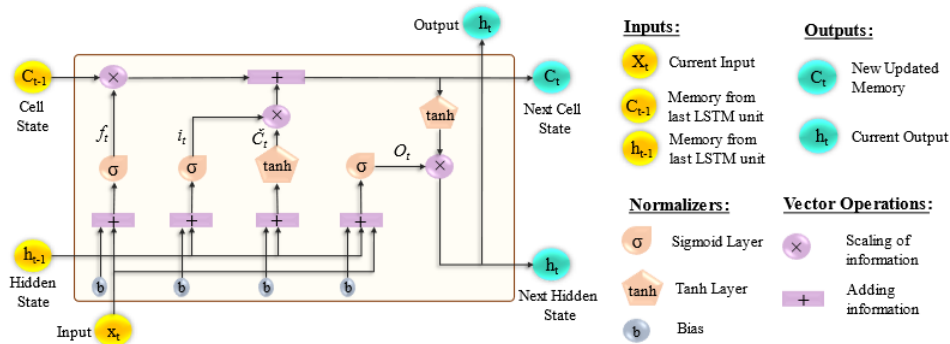


Figure 3. The structure of the LSTM neural network

The first stage in the LSTM decides the discarded information from the cell state. The Sigmoid layer makes this judgment and acts as the “forget gate”. Forget gate takes the h_{t-1} and x_t values, examines them, and returns a number between 0 and 1 for each cell state C_{t-1} , in this case, ‘1’ indicates “keep all” and ‘0’ indicates “delete all”. The next stage having two elements (a Tanh and a Sigmoid), examines the new data that has been saved in the neural network unit. The Tanh function provides a vector \tilde{C}_t with new candidate values inserted after a Sigmoid layer termed as “input” gate which determines the values to be modified (i_t). As shown in (1) and (2) illustrate the computation, where W_f and W_i are the matrices of the weights of the input gate and input elements, and b_f and b_i are the associated biases. The third stage serves to switch the state of the cell from old (C_{t-1}) to new (C_t). As shown in (4) is the computation equation and specifies which information from the previous unit status is saved and determines the value of a new candidate. The output of the LSTM is determined according to the filtered cell state. Finally, the output of both layers is multiplied, and hence, only the chosen parts are retrieved as the output. As shown in (5) and (6) illustrate the computation process.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (1)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (3)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

In this study, a neural network model with five layers is constructed for sugarcane crop identification using temporal RS data as shown in Figure 4. The model comprises an input layer, LSTM layer(s), two dense layers with 256 and 128 nodes, and an output layer (Softmax layer). The input to the network is the spectro-temporal reflectance of the crops stacked with NDVI during the main sugarcane growth stages whereas, the output of the model is the label for each pixel. Dropout is used to prevent overfitting, keeping the dropout rate to 10 percent for both dense layers. The model performance is assessed with 1, 2, and 3 LSTM layers with 256 hidden nodes each. For optimizing network parameters, different optimizers are used with a learning rate of 0.001. The architecture is also tested for different batch sizes.

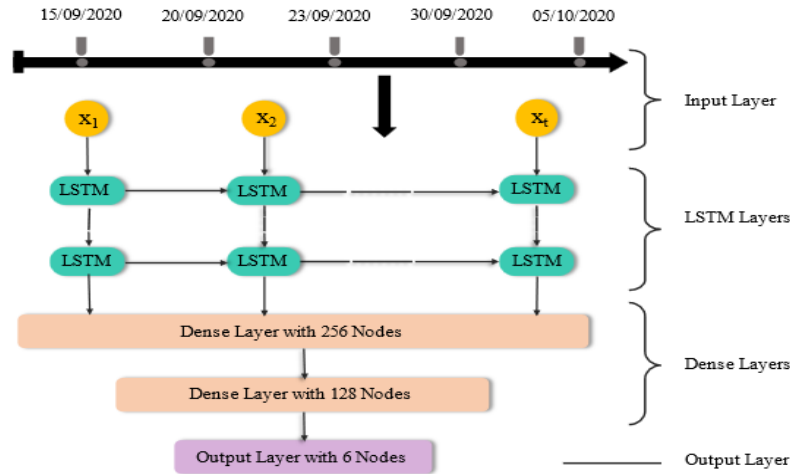


Figure 4. The architecture of the LSTM model for crops identification

2.4.2. Grid search hyperparameter optimization

The basic concept of grid search is to explore all potential parameter configurations. The tuning and optimization of N number of parameters of the DL model require building an N -dimensional grid. The N -dimensional grid can be represented as $G = (p_1, p_2 \dots p_N)$, where p_i represents the i^{th} parameter that needs to be tuned. Furthermore, each dimension can be represented as, $([p_{11}, p_{12} \dots p_{1m}], [p_{21}, p_{22} \dots p_{2n}] \dots [p_{N1}, p_{N2} \dots p_{Nt}])$ where m , n , and t are the number of possible values of parameter p_i . The grid search requires $m * n * \dots * t$ iterations in order to be completed. When the grid's dimensions are large, a great number of computational resources are required, making the search more challenging. On the other hand, when dimensions are small, the grid search optimization approach is more favorable. The grid in this study comprises four dimensions which makes grid searching simple and easy. The total number of parameter sets is $3 * 7 * 7 * 7 = 1,029$ wherein 3 layers, 7 activation functions, 7 batch sizes, and 7 optimizers are evaluated as explained in section 2.4.3.

2.4.3. Evaluated hyperparameters

To minimize the computational cost, a limited range of different hyperparameters were considered (discussed below) during grid search optimization of LSTM architecture. The approach used in this study relies on a database to store the experimental history of configurations that have been tried and the values of loss and accuracy for each hyperparameter set. As the search progresses, the database grows and the algorithm explores the complete search space. The hyperparameters evaluated in this study are mentioned in the rest of this section.

a. Optimizer

The optimizer is responsible for the neural network's objective function minimization. Stochastic gradient descent (SGD) is a popular optimizer for the efficient and successful optimization of machine learning techniques [14]. SGD may be strongly influenced by the value of the learning rate chosen. There are many other gradient-based optimization algorithms proposed. The optimizers evaluated in this study are AdaGrad [15], Adadelata [16], Adamax [14], root mean square propagation (RMSProp) [17], Adam [14], and Nadam, an Adam variant that incorporates Nesterov momentum [18]. The outcomes are discussed in section 3.

b. Number of LSTM-layers

Deep neural networks are structured in a hierarchical manner, with each level becoming more complex and abstract, unlike conventional linear machine learning techniques. Each layer performs a nonlinear conversion on its input and, using what it has learned, outputs a statistical model. Multi-layer deep neural networks are more efficient compared to one or two-layer deep networks [19]. There has been no exact method until today to find out how many layers will be enough for the network instead the common practice is to use a trial-and-error method is used to find the depth of the network giving the best results. In this study, we evaluated the model by keeping the number of dense layers to 2 and varying the LSTM layers to 1, 2, and 3 layers. The outcomes are discussed in section 3.

c. Batch size

Gradient update for each training sample and gradient update on the parameters from the complete training set are the two extremes that can be used to optimize the weights of a neural network. Mini batch gradient descent is the middle way where the parameters of the network are updated for a small sample (batch) of the training set. This study evaluated mini-batch sizes of 16, 32, 64, 128, 256, 512, and 1024 and results are reported in section 3.

d. Activation function

Different activation functions can be used in different layers of the model and they have a substantial influence on the capabilities and performance of a neural network. The output layer of a multi-class classification model typically uses the Softmax activation function, while the hidden layers of a neural network use a differentiable nonlinear activation function. This allows the model to learn more complex functions than those trained with a linear activation function. Some of the commonly used activation functions are rectified linear unit (ReLU) [20], scaled exponential linear units (SELU) [21], exponential linear units (ELU) [22], Softsign [23], and Softplus [24]. The effect of different activation functions on the performance of the LSTM model used in this study for crop classification can be found in section 3.

2.4.4. Loss function: categorical cross-entropy loss

Generally, to train any neural network, an objective is needed to measure the performance of the network and adjust the weights of the network during training. It computes the mean difference between the actual and predicted probability distributions for all classes in the given problem. The loss function considered for this study is categorical cross entropy which is commonly used for classification problems and yields better results compared to other loss functions [25]. Categorical cross-entropy cost function $CCE(X, w, b)$ with given input x_m , M inputs with $m = \{1, \dots, M\}$, weights w , and biases b , K target binary output variables t_k , with $k = \{1, \dots, K\}$, the error function is shown in (7) with the predicted output variables $y_{mk}(x_m, w, b)$ for input m and class k .

$$CCE(X, w, b) = -\sum_m^M \sum_k^K t_{mk} \ln y_{mk}(x_m, w, b) \quad (7)$$

3. RESULTS AND DISCUSSION

The network with all the hyperparameter sets is run three times and the performance is measured using the average of these runs. For all the configurations, the percentage of training and testing data is set at 70%, and 30% respectively. The model was trained for a total of 25 iterations. In the rest of this section, firstly sensitivity analysis is carried out to study the effects of four hyperparameters (explained in section 2.4.3.) and different combinations of these hyperparameters are tried to see their effects on the results. Consequently, to highlight the individual effects of hyperparameters, an effective illustration of the results is given in section 3.2.

3.1. Effect of hyperparameter combinations

The resultant performance of the LSTM model using different hyperparameter combinations is shown in Figures 5(a) and (b). Figure 5(a) shows the minimum loss of the model and Figure 5(b) shows the average loss for all the tested batch sizes. In regards to optimizers, it can be seen that irrespective of other hyperparameters, RMSProp, Adam, Nadam, and Adamax optimizers achieved excellent performance, whereas, Adagrad and Adadelata showed poor performance. SGD showed good performance in some cases but had a mediocre average performance overall, and it decreased the model performance when the number of layers was increased to 3, unlike other optimizers that have either the same performance as lesser layers or increased performance. The model performs worst when Adadelata is used in combination with Sigmoid irrespective of the number of layers and batch sizes. The best set of hyperparameters giving the minimum loss compared to all 1,029 possible combinations of hyperparameters, and the minimum average loss of all the combinations found with the experimentation is presented in Table 2.

3.2. Effect of individual hyperparameter

The optimizers evaluated in this study are SGD, Adagrad, Adamax, Adadelata, RMSProp, Adam, and Nadam. For all the optimizers, we used the default value of 0.001 for the learning rate. Figures 6(a) and (b) show the performance of different choices of optimizers for the crop identification task. It is observed that the best performance is achieved using the Nadam optimizer yielding the least loss on the tested configurations whereas the minimum average loss is achieved using the Adam optimizer. It is observed that the average loss for RMSProp, Adam, Nadam, and RMSProp is much smaller in comparison to SGD, Adagrad, and Adadelata. It would be interesting to evaluate other hyperparameters of these optimizers instead of using the default values and observe how they affect the model's performance. The activation functions evaluated in this study are ELU, ReLU, SELU, Tanh, Softplus, Softsign, and Sigmoid. Figures 7(a) and (b) show the performance of the choices of activation functions for the crop identification task. It is observed that the best performance was achieved using Tanh yielding the least loss on the tested configurations whereas the least average loss is achieved by using Elu as an activation function. It can be observed that the average loss for ELU, ReLU, SELU, and Tanh is much smaller in comparison to Sigmoid, Softplus, and Softsign. The model performs worst when Sigmoid is used as the activation function.

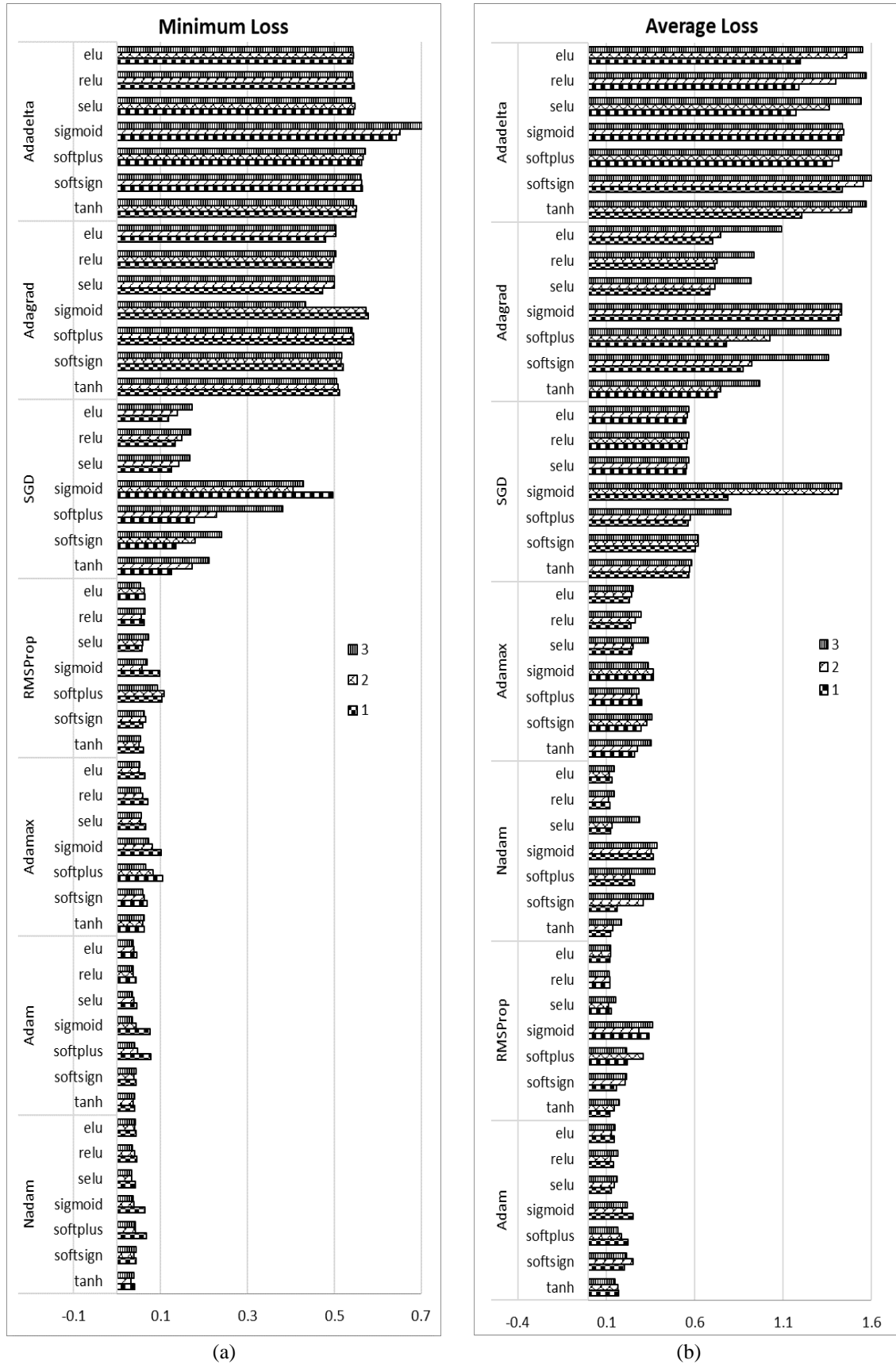


Figure 5. Effect of different combinations of hyperparameter sets on the performance of LSTM model using 1,2 and 3 layers (a) minimum loss (b) the average loss

Table 2. Hyperparameter sets with the best performance for the crop identification task

| Hyperparameters | Best set giving minimum loss | Best set giving minimum average loss |
|---------------------|------------------------------|--------------------------------------|
| Optimizer | Nadam | Adam |
| Activation Function | Tanh | ELU |
| Batch Size | 32 | 16 |
| Number of Layers | 2 | 1 |

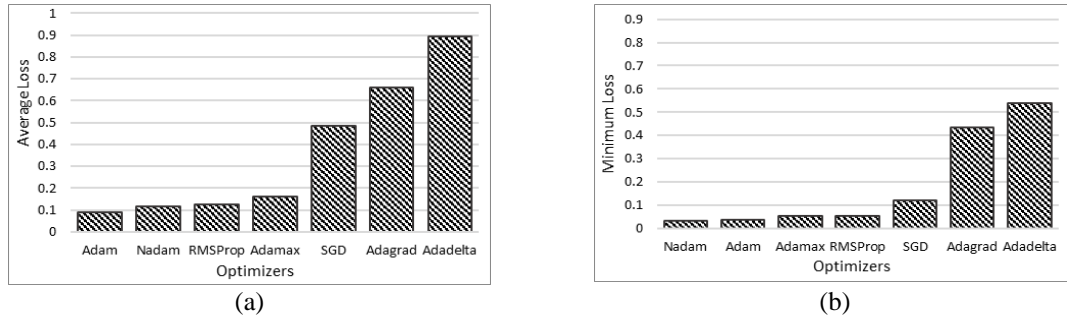


Figure 6. Effect of choice of optimizer on the performance of LSTM model (a) average loss and (b) minimum loss

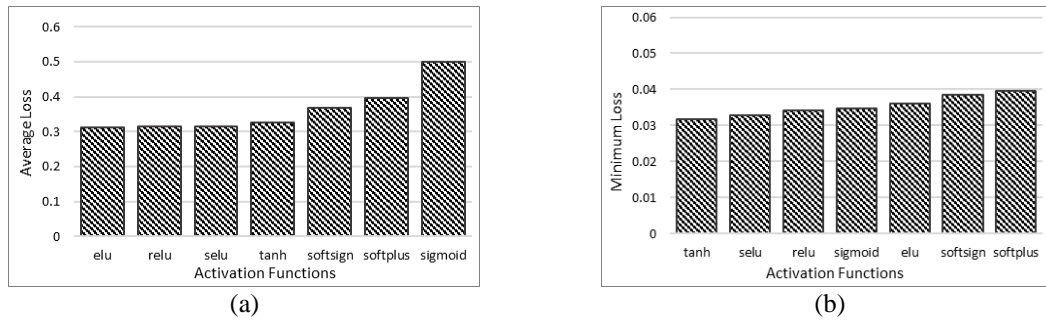


Figure 7. Effect of activation function on the performance of LSTM model (a) average loss and (b) minimum loss

The model is tested for 1, 2, and 3 LSTM layers. The performance of the LSTM model for a different number of layers can be seen in Figures 8(a) and (b). It is observed that the model performs best with one or two layers in most cases. An increase in the number of layers not only increases the training time but also degrades the performance since it magnifies the model’s loss. The model is tested for 7 different mini-batch sizes i.e., 16, 32, 64, 128, 256, 512, and 1,024. It can be seen in Figures 9(a) and (b) that as we increase the batch size, the average loss of the model also increases, and the overall minimum loss was seen with the batch size of 32 using the Nadam optimizer with the Tanh activation function.

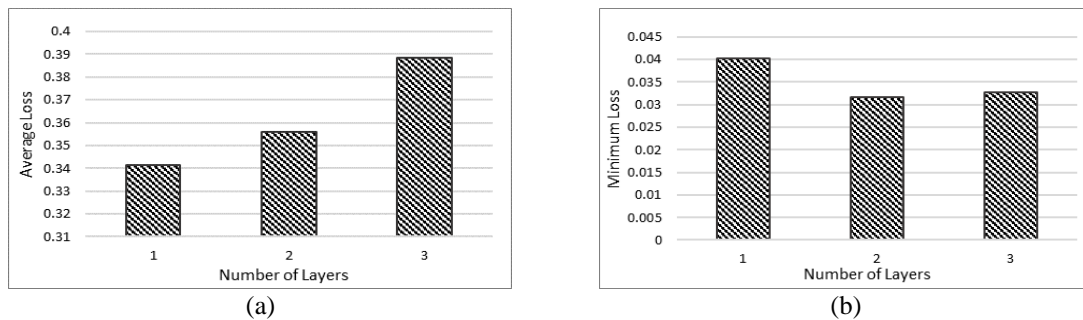


Figure 8. Effect of choice of the number of layers on the performance of LSTM model (a) average loss and (b) minimum loss

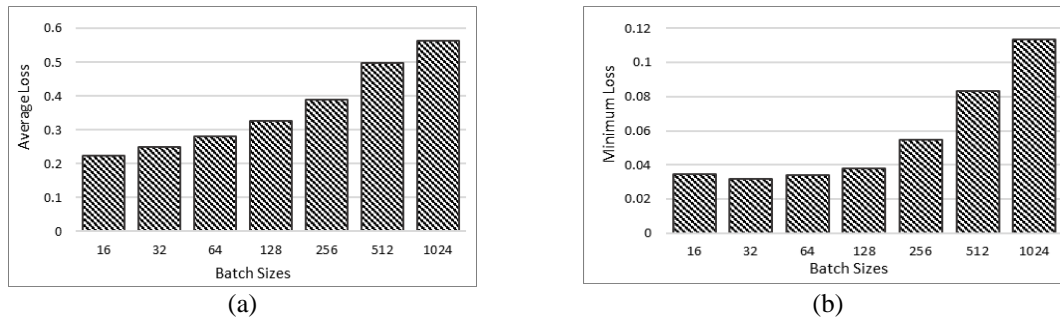


Figure 9. Effect of choice of batch size on the performance of LSTM model (a) average loss and (b) minimum loss

4. CONCLUSION

This study shows that tuning the hyperparameters improves the model performance. In this study, the result of the grid search was analyzed to find the effects of four hyperparameters, namely optimizers, activation functions, number of layers, and batch size, on the performance of the LSTM model. Grid search tested the model for a total of 1,029 sets of four hyperparameters and the results for all the sets stored in a database were analyzed to see their influence on the effectiveness of the model. The LSTM model in this study was tested for 1, 2, and 3 number of hidden LSTM layers, and results showed that the number of layers did not make any significant difference to the performance of the model. It is also concluded from the results that the LSTM model for RS data yields the best performance with Adam, Nadam, RMSProp, and Adamax optimizers whereas it does not perform well with SGD, Adagrad, and Adadelta. The results show that the choice of activation function also influenced the effectiveness of the model. It is further concluded that the performance of the model decreases with the increase in the batch size and the best value for the batch size is 16 or 32 depending upon which other hyperparameters are used with it. Increasing the batch size significantly reduces the training time but limits the performance of the LSTM model.




REFERENCES

- [1] S. Foerster, K. Kaden, M. Foerster, and S. Itzerott, "Crop type mapping using spectral-temporal profiles and phenological information," *Computers and Electronics in Agriculture*, vol. 89, pp. 30–40, Nov. 2012, doi: 10.1016/j.compag.2012.07.015.
- [2] E. Ndikumana, D. H. T. Minh, N. Baghdadi, D. Courault, and L. Hossard, "Deep recurrent neural network for agricultural classification using multitemporal SAR sentinel-1 for camargue, France," *Remote Sensing*, vol. 10, no. 8, Aug. 2018, doi: 10.3390/rs10081217.
- [3] H. C. de C. Filho *et al.*, "Rice crop detection using LSTM, Bi-LSTM, and machine learning models from sentinel-1 time series," *Remote Sensing*, vol. 12, no. 16, Aug. 2020, doi: 10.3390/rs12162655.
- [4] C. Pelletier, G. Webb, and F. Petitjean, "Temporal convolutional neural network for the classification of satellite image time series," *Remote Sensing*, vol. 11, no. 5, Mar. 2019, doi: 10.3390/rs11050523.
- [5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [6] H. Jiang *et al.*, "A deep learning approach to conflating heterogeneous geospatial data for corn yield estimation: A case study of the US Corn Belt at the county level," *Global Change Biology*, vol. 26, no. 3, pp. 1754–1766, Mar. 2020, doi: 10.1111/gcb.14885.
- [7] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.
- [8] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, and S.-H. Deng, "Hyperparameter optimization for machine learning models based on bayesian optimization," *Journal of Electronic Science and Technology*, vol. 17, no. 1, pp. 26–40, 2019, doi: 10.11989/JEST.1674-862X.80904120.
- [9] I. Priyadarshini and C. Cotton, "A novel LSTM-CNN-grid search-based deep neural network for sentiment analysis," *The Journal of Supercomputing*, vol. 77, no. 12, pp. 13911–13932, Dec. 2021, doi: 10.1007/s11227-021-03838-w.
- [10] D. P. Roy, H. Huang, R. Houborg, and V. S. Martins, "A global analysis of the temporal availability of PlanetScope high spatial resolution multi-spectral imagery," *Remote Sensing of Environment*, vol. 264, Oct. 2021, doi: 10.1016/j.rse.2021.112586.
- [11] M. Sudmanns, D. Tiede, H. Augustin, and S. Lang, "Assessing global Sentinel-2 coverage dynamics and data availability for operational Earth observation (EO) applications using the EO-Compass," *International Journal of Digital Earth*, vol. 13, no. 7, pp. 768–784, Jul. 2020, doi: 10.1080/17538947.2019.1572799.
- [12] N. Minallah, M. Tariq, N. Aziz, W. Khan, A. ur Rehman, and S. B. Belhaoui, "On the performance of fusion based planet-scope and sentinel-2 data for crop classification using inception inspired deep convolutional neural network," *PLOS ONE*, vol. 15, no. 9, Sep. 2020, doi: 10.1371/journal.pone.0239746.
- [13] P. Dimitrov *et al.*, "Sub-pixel crop type classification using PROBA-V 100 m NDVI time series and Reference data from sentinel-2 classifications," *Remote Sensing*, vol. 11, no. 11, Jun. 2019, doi: 10.3390/rs11111370.
- [14] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *Prepr. arXiv1412.6980*, Dec. 2014.
- [15] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.
- [16] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," *Prepr. arXiv1212.5701*, Dec. 2012.




- [17] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Cited on*, vol. 14, no. 8, pp. 2–31, 2012.
- [18] T. Dozat, "Incorporating second-order functional knowledge for better option pricing," in *Workshop track - ICLR 2016*, 2016, pp. 1–4.
- [19] C. Baral, O. Fuentes, and V. Kreinovich, "Why deep neural networks: A possible theoretical explanation," in *Constraint Programming and Decision Making: Theory and Applications*, 2018, pp. 1–5. doi: 10.1007/978-3-319-61753-4_1.
- [20] A. F. Agarap, "Deep Learning using rectified linear units (ReLU)," *arXiv preprint arXiv:1803.08375*, Mar. 2018.
- [21] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 972–981.
- [22] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," *Prepr. arXiv1511.07289*, Nov. 2015.
- [23] J. Turian, J. Bergstra, and Y. Bengio, "Quadratic features and deep architectures for chunking," in *Proceedings of NAACL HLT 2009*, 2009, pp. 245–248.
- [24] C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, and R. Garcia, "Incorporating second-order functional knowledge for better option pricing," in *Advances in Neural Information Processing Systems 13 (NIPS 2000)*, 2000.
- [25] C. M. Bishop, *Pattern recognition and machine learning*. Springer New York, NY, 2006.

BIOGRAPHIES OF AUTHORS






Madiha Sher    is working as a Lecturer at the Department of Computer Systems Engineering, University of Engineering and Technology, Peshawar, where she obtained her B.Sc. and M.Sc. degrees in Computer Systems Engineering. She was awarded a gold medal in recognition of her first-place finish in the B.Sc. program. She is currently pursuing her Ph.D. from the same department. Her research interests span various topics, including machine learning, deep learning, and remote sensing. She is passionate about exploring new avenues for improving agriculture practices in Pakistan. She can be contacted at email: madiha@uetpeshawar.edu.pk.






Nasru Minallah    graduated with his Ph.D. from Southampton University, UK in 2010 with a focus on Multimedia and its applications. Dr. Nasru did his MSc from LUMS in 2006 and his BSc in computer systems engineering from the University of Engineering and Technology (UET), Peshawar, Pakistan in 2006. Furthermore, he has been a postdoctoral fellow in France. He has published his work in several IEEE conferences and journals. He is PI and Co-PI on several research grants. Email: n.minallah@uetpeshawar.edu.pk.



Tufail Ahmad    is working as a research associate at National Center for Big Data and Cloud Computing (NCBC). He received his B.Sc. degree from Agriculture University Peshawar, in Computer Sciences, attaining a gold medal. He is currently pursuing his M.Sc. in Computer Science from the University of Engineering and Technology Peshawar. He can be contacted at email: tahmad@uetpeshawar.edu.pk.



Waleed Khan    received his B.Sc. degree in Electrical Computer Engineering from Comsats University Islamabad, Abbottabad Campus Pakistan in 2015. The MSc Degree in Computer Systems Engineering from the University of Engineering and Technology, Peshawar (UET Peshawar), Pakistan. He is pursuing his Ph.D. from the Department of Computer Systems Engineering at UET Peshawar. His main area of research involves deep learning algorithms in Remote Sensing. He can be contacted at email: khanwaleed247@uetpeshawar.edu.pk.