Efficient DNA Mutation Classification Using Deep Active Learning Techniques

Rana H. Saloom[1], Hussein K. Khafaji[2], Nabaa Ali Abd Alradha Al-saeedi[3], Saif Ali Abd Alradha Alsaidi[4] and Furat Al Daamee[5]

[1]*Private University Education Department, Ministry of Higher Education and Scientific Research, 10011 Baghdad, Iraq*
[2]*Computer Communications Engineering Department, Al-Rafidain University College, 10011 Baghdad, Iraq*
[3]*Department of Pathological Analysis, College of Science, Wasit University, 52001 Kut, Wasit, Iraq*
[4]*College of Computer Science and Information Technology, Wasit University, 52001 Kut, Wasit, Iraq*
[5]*University of Technology and Applied Sciences Al Mussanah, 312 Al Mussanah, Oman*
Ranasaloom3081978@gmail.com, Hussain.ketan.elc@ruc.edu.iq, Fatemahhameed1984@gmai.com, salsaidi@uowasit.edu.iq, Furat.abdulhussain@utas.edu.om

Keywords: Mutations, Variants, Active Learning, Genetic Classification, DNA.

Abstract: Deep neural networks have shown impressive performance in a variety of fields, but the challenge in the training procedure for these networks may be time-consuming, particularly when working with huge datasets. Active learning is a potentially useful method that may speed up the training process by prioritizing the selection of samples that include the most relevant information for annotation. In this study, we propose an architecture for deep neural networks that combines active learning with Long Short-Term Memory (LSTM) in order to improve the effectiveness of the training process for DNA mutation categorization which is our contribution. We test the suggested method on the Cancer Cell Lines Encyclopedia (CCLE) dataset and assess its performance in comparison to a deep network and an LSTM network that do not use active learning. According to our experiments, the proposed approach is more accurate and improves efficiency by significantly reducing training time compared to other methods. When comparing deep learning models with deep active models, we found that the former averages a much higher training time of 4859.678 seconds while the latter averaged a substantially lower first training epoch of 978.8522 seconds, making total prediction and training time approximately 1474.385 seconds for the entire first phase. Focusing on some aspects outside of model creation as well, our research demonstrates the promise active learning holds in speeding up the training process for deep networks in context of classifying Deoxyribonucleic acid DNA mutations and emphasizes important aspects needed in constructing effective deep learning models.

1 INTRODUCTION

Deep learning has significantly transformed the areas of artificial intelligence study, and has been successfully applied in many sectors such as image recognition, natural language processing, and voice recognition [1]. Improve the training performance is the main challenge in deep learning that we try to fix it. Performance in different tasks has been improved with the use of deep neural networks; for example, deep learning is helping to enhance genomic research through the classification of DNA mutations [2]. In the classification process of DNA mutations, detecting changes in DNA sequences that could be associated with diseases like cancer is essential [3]. Training deep neural networks for classifying DNA mutation types may be especially slow when dealing with large datasets containing thousands of samples. This is particularly true given that there are various methods for classification [4] - [6].

Active learning is a useful technique that could potentially reduce the amount of time needed to train deep neural networks by selecting the most pertinent information to annotate [7], [8]. Its application is well-known in many areas including natural language processing, computer vision, and bioinformatics, among others. Active learning relies on the principle of iterative selection where a subset is chosen from an unlabeled dataset and labeled by a domain expert—like a biologist or doctor—and then the model is updated using those labeled examples [9], [10]. This process continues until the model reaches a target accuracy threshold set as “good enough.”

In this research, we are developing the structure of a deep neural network that uses active learning combined with long short-term memory (LSTM) to

make the training process more efficient for classifying DNA mutations. The recurrent neural network known as LSTM has the ability to recognize temporal relationships in a succession of inputs. LSTMs have been proven useful in an array of tasks that involve sequences, such as processing language, recognizing speech, and analyzing video footage. These are some examples of fields where LSTMs have been applied. We speculate that applying LSTM in deep neural networks may enhance the precision for classifying DNA mutations by improving the recognition of chronological dependencies found within DNA sequences. The dataset from the Cancer Cell Line Encyclopedia (CCLE) serves as an example to test our model. It contains genomic information of over a thousand cancer cell lines and is publicly available, which is useful to us for evaluating the success of our method. We assess our technique against a benchmark deep neural network that does not incorporate active learning. The results from our experiments confirm that this approach achieves better accuracy and significantly reduces training time compared to other methods. As a whole, this research highlights the potential that active learning offers for speeding up the training process of deep neural networks with STM in the context of DNA mutation classification. The system we designed could reduce expenditures in both time and cost for DNA mutation classification. Moreover, it provides insight into constructing advanced models of deep learning intended for bioinformatics applications.

2 RELATED WORKS

Now, let’s look at the advancements made on the matter of shortening deep network training time and focus on some of its research aspects.

In 2016, Viacheslav Khomenko et.al. presented a both effective and efficient algorithm for his recurrent neural networks rained over minimum spanning trees. The approach that has been recommended takes into account optimal batch bucketing based on the input sequence length and the subsequent distribution of data across many GPUs. The initial training baseline with no sequence bucketing is benchmarked against for multiple varying numbers of buckets after employing sequence bucket scheduling [11]. During evaluation, wall clock time, epochs, and validation loss were all optimally tuned to achieve desired results while retaining computational efficiency [12].

In 2018, Guojing Cong and others proposed new methods of speeding up training, including algorithms. They suggested a distributed method that does better than most asynchronous stochastic gradient descent in both convergence and scalability. It has an adaptable batch size and a distributed training environment. It reduces the communication costs and decreases the number of iterations required for convergence. The authors optimize the Adam algorithm to fit their distribution so as to increase the program's efficiency in total. Other strategies include transfer learning, which increases accuracy during validation while speeding up training time [11], [13].

DAPP is shorthand for Distributed Acceleration of Training of Deep Neural Networks Using the Ping-Pong Approach. This acceleration strategy was introduced by Sapna et al. in 2018, aiming to reduce the time required for training deep neural networks (DNNs). The application of DAPP is helpful for DNNs with numerous computationally intensive layers because it offers a solution towards speeding up DNN training. DAPP uses model parallelism together with local distributed memory to improve training speed without degrading the model quality. Since different levels computed gradient within tiers are processed in parallel and applied to advance blocks of data within a pipelined structure, no global memory is necessary [14], [15].

A study conducted in 2018 by Naisen Yang and others explores the effectiveness of Stochastic Gradient Descent (SGD) and other adaptive optimization methods like RMSprop and Adam in training deep neural networks. The mentioned methods apply to cases where an entire dataset can be used during each epoch, but large datasets often have duplicated or irrelevant data. To address this issue, the authors propose a new method which they call DropSample. This method reduces the time it takes to train Convolutional Neural Networks (CNNs) by intentionally saving fewer samples or as they put it "sample reduction." One way of understanding DropSample is as an approach to truncated cross-entropy loss with limited margins [16], [17].

A study performed by Chih-Chieh Yang and Guojing Cong in 2019 looked at the impact data loading has on the time it takes to train deep neural networks in large scale distributed systems. They specifically look at ways to improve data loading speeds for these situations. While designing data loaders, they propose some Center Processing Unit CPU resource optimizations based during the design sprints. Using analytical models, they carry out a controlled investigation on how data loading impacts training times and perform trend analyses over larger dispersed training scales. The model confirms that distributed training scalability was being hindered by an I/O speed bottleneck, thus motivating the need for

context-aware local area data loading methods. Unlike first approaches, this method significantly reduces both transient data transfer volumes as well as permanent storage cache hits due to software caching techniques [18].

3 ACTIVE LEARNING

Active learning is a type of ROMN, issue model which utilizes freshly labeled data points. The data points to be worked on are continuously renewed in this form of learning. It is typical to use the active learning approach in cases where there exists a large amount of unlabeled with minimal labeled annotations, Actional USA human or machine is costs [19]. The main principle on which active learning rests focuses on heuristic sample selection. Through its architecture, models acting under reinforcement paradigms can pose dleon their own seek an oracle's answer hmn annotator) for selected queries.

This is different from the approach of waiting for labeled data to come in, which is often referred to as “sitting back and relaxing”. Active learning focuses on improving a model while minimizing labeled data through instance selection. Labeled data in active learning employs many diverse methods around labeling example data. As two broad classifications query tactics and model-based strategies could be used:

Query Strategies. Based upon heterogeneity or randomness of the data. Query take into consideration challenge posing instances for labeling. The following are some examples of frequent query strategies:

Uncertainty sampling selects cases computational models predict with most ambiguity internally. Further, it may be divided more than few ways like high entropy or low margin selection. Diversity Sampling wants to choose examples from different segments of data in order to provide the model with complete training [20].

Representative Sampling ensures that important portions of the data are adequately represented or captured by selecting examples out of a dataset that they consider typical of and proportional to the entire dataset [21].

Model Based Strategies. These strategies utilize the model’s inner structure and decision boundaries for guiding instance selection. Often, these methodologies include steps where the predicted model is evaluated on unlabeled instances to control ranking of labeling stages. A few examples of such strategies are outlined in [22].

Bayesian Active Learning. Opt for samples that have the highest posterior entropy, then estimate the level with which one is uncertain using Bayesian inference [23].

Query-by-Committee. Maintain an ensemble of models; classifying samples that would invoke maximum disagreement or ambiguity amongst the ensemble members [24].

Anticipated Model Change. This technique tries to anticipate how the performance of a model will change in relation to labeling an instance [25].

Considering all of the active learning methods available, one must choose the most appropriate based on the qualities of data, problem domain and issue at hand, and other available resources. Some techniques may perform better than others depending on the specific controlled task and dataset characteristics. Additionally, different methods may exhibit varying levels of performance.

4 METHODOLOGY

The categorization of DNA mutations is an extremely important step in both comprehending the genetic underpinnings of illnesses and developing individualized treatment strategies. However, in order to correctly categorize DNA mutations, a large quantity of labeled data is often necessary. Acquiring this data may be difficult and costly owing to the time-consuming process of expert annotation that is required. Active learning is a potential strategy to solve this problem since it allows for the deliberate selection of the most informative DNA variants for labelling. As a result, the annotation effort may be reduced while classification performance might be maintained or improved. We followed the steps in Figure 1 when classifying DNA mutations using active learning.

4.1 Dataset Preparation

We obtain a dataset of DNA mutations with associated labels indicating their classification from DepMap[1], which hosts one of the mutation data sets that may be accessed by researchers. It is a gateway for Cancer Cell Lines (also known as CCLE). It offers the scientific community a forum in which to exchange information and ideas about cancer susceptibilities. The Mutation Annotation Format file (MAF) is what this data collection is comprised of. This file contains information on all of the somatic

[1]https://depmap.org/portal/#.

point mutations and indels that were found in the DepMap cell lines. The document stores a data collection that is comprised of 1,048,575 rows, each of which represents a particular kind of genetic mutation. We used this dataset in our work on DNA mutation classification.

Then we create three subsets from the dataset: an initial labelled set consisting of 20% of the dataset, an unlabelled pool consisting of 60% of the dataset and 20% represent the validation data.

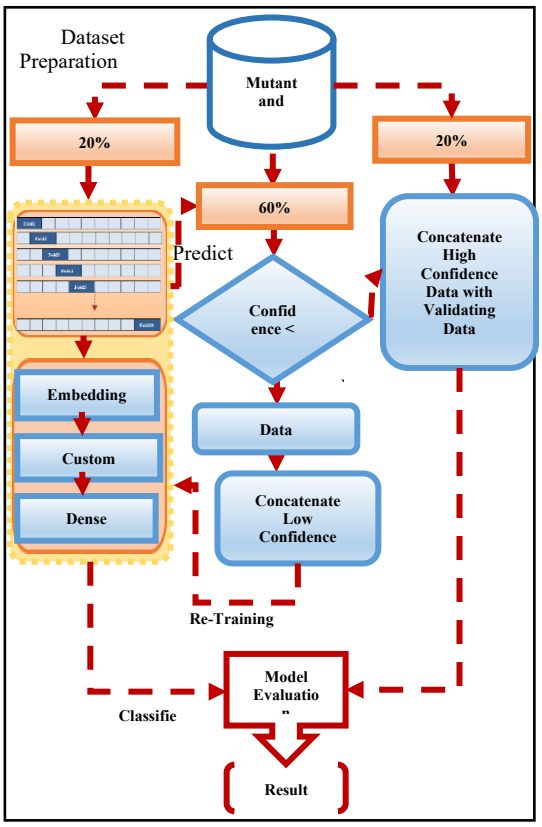

Figure 1: Deep-Active learning steps.

4.2 Model Initialization and Training

In the process of building the model, an LSTM technique was used, and a variety of neural network topologies were tested and compared against one another to see which one delivered the greatest results.

During the phase of the model when it is being trained, the network receives the reference codon as well as the query codon as input. In a neural network, one of the hidden layers is called an embedding layer. This layer takes data input from a high-dimensional domain and translates it to a lower-dimensional domain. This helps the network learn about the links between the input data and process it more efficiently. The use of LSTM networks in this setting could prove advantageous in the long run. One of the characteristics that sets LSTM apart from other similar algorithms is its capacity to preserve data sequences. It gets rid of information that isn't being used, and as we all know, data is always full of a lot of information that isn't being used. This is because it gets rid of information that isn't being used. This information may be omitted from the computation when using LSTM, which can both speed things up and save expenses. Due to this capability, LSTM is an effective method for classification since it can eliminate unnecessary information while keeping the information sequence intact. Both the reference codon and the query codon are input into the network. An LSTM cell's inputs are the hidden state vector and cell state vector from the preceding cell. Our final result is a merged hidden and cell-state vector. Cells in the next generation are fed the hidden state and cell-state vectors from the cell before them, as well as the previous generation's model input. Once those two vectors have been updated, they are sent to the next cell, and the concealed state vector is passed on to the dense layer, which is the output layer. A posterior distribution across the next generation word at that position is then output by the dense layer. Applying a SoftMax function to the data yields a probability distribution. Figure 2 shows the architecture of the proposed model.

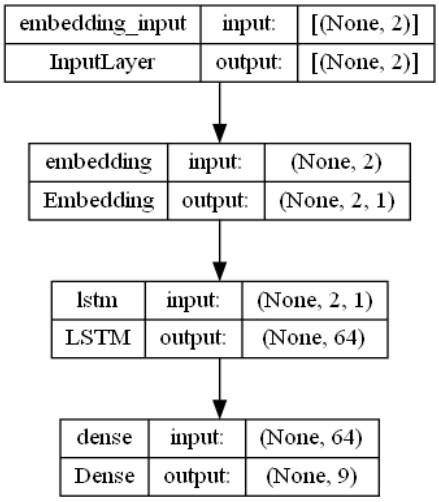

Figure 2: The proposed model architecture.

Following the construction of a model, the next critical stage in the process of deep learning is to train the model using data that has been labeled (20% from the original dataset). Because of this process, the

model is able to learn patterns, correlations, and features from the dataset that is presented to it. This gives it the ability to properly make predictions or categorize new instances that have not been seen before. During the training phase, the model uses these labels as the ground truth, which enables it to comprehend the required mapping between the input data and the desired output.

4.3 Selection of Active Learning Strategy

When deciding which active learning strategy to use, it is important to take into consideration both the specific tasks that need to be completed and the resources that are available. Therefore, we pick an efficient active learning method that has the potential to be used for the classification of DNA mutations as uncertainty sampling. In uncertainty sampling, examples for labeling are chosen based on whether or not the model considers them to be uncertain or ambiguous. Taking into account the model's anticipated probability or confidence ratings is one way to accomplish this goal. Instances with a high degree of prediction uncertainty are chosen for labeling because they have the ability to provide the most useful input to the model and may assist in the process of lowering the model's overall level of uncertainty. This was accomplished by first applying the prediction to the testing data, which represents 60% of the original dataset without the labels. Next, we chose the mutation samples that have a high degree of uncertainty in the model (samples whose predicted probability is less than 0.6).

4.4 Model Retraining

The process of selecting instances with high prediction uncertainty is typically done using uncertainty sampling, which evaluates the confidence or uncertainty of the model's predictions based on the data that resulted from the previous step and its label from the original dataset. Model retraining is an essential step in the active learning process. Following the selection of these unsure cases for classification, they are combined with the 20% of the initial data that had already been classified. The training set for the process of retraining the model is going to be this merged dataset, which will consist of both the original labeled data and the newly labeled cases. When the model is retrained, it is given an updated version of the larger training set to use. In order to achieve the goal of minimizing the error or loss function, the internal parameters or weights of the model are modified via an iterative optimization process. Through the process of retraining, the model is given the opportunity to gain knowledge from freshly labeled examples, enabling it to incorporate new information and enhance its capacity for prediction.

The model concentrates its attention on parts of the data distribution where it had trouble making accurate predictions in the beginning. This is expected to assist the model in easing its grasp on intricate patterns, challenges, or sparse areas with scant labeled data. Also, it is forecasted that these examples will provide helpful insights. Reduction of prediction uncertainty is sought to enhance how well a retrieved model performs on unseen data and improve its overall performance. The active learning pipeline continuously selects uncertain cases and trains the model afresh using new labels and information which improves the model’s accuracy and ability to generalize.

5 ASSESSING PERFORMANCE

After performing sequential training and retraining on the model using active learning, assessing performance measuring define generalization becomes pivotal. The trained model’s generalization capability can also be tested. This evaluation is normally done withheld-out test sets which consist of twenty percent of the original(validation) dataset plus certain samples whose predicted probabilities are repeatively above 0.6 threshold. The method of evaluating the model entails introducing the test data into the trained model and contrasting its predictions with the labels that are known to represent the ground truth. Using Accuracy, Precision. Recall, F1 Score

Then we analyze the amount of time needed to train a model with active learning vs. a model without active learning, and we investigate all characteristics of both situations that are connected to the concept of time. When a model does not incorporate active learning, the training stage typically involves using the entire labeled dataset. This approach can be quite time-consuming, especially with large datasets. To fine-tune its parameters and reduce the loss function, a model goes through multiple iterations to increase accuracy. The time it takes to train a model is directly connected to how complicated the model is, the amount of data and computing hardware available.

In contrast to models without active learning, the use of active learning may significantly reduce the time efficiency of training. Active learning includes selecting cases for labeling in an example-specific

and controlled way, focusing exclusively on those cases that are expected to have a maximum impact on model performance. By using instance selection based on label information, it is possible to achieve equivalent performance to a model without active learning with very few labeled examples. This drop in training data volume could enable improved convergence during the training process which ultimately results in reduced time requirements for training.

6 RESULTS

In this work, we investigate the potential applications of active learning in combination with deep learning models for the categorization of DNA mutations. The purpose of this research is to determine if active learning may successfully shorten the amount of time spent training while still retaining classification performance on par with that of more conventional deep learning methodologies. Our goal is to accelerate the learning process of the model while simultaneously minimizing the amount of exhaustive annotation that is required by using a methodical approach to choosing the DNA mutations that will be labelled as informative.

Using the CCLE dataset, we performed tests on a DNA mutation classification problem using a deep learning architecture that was particularly created for this purpose in order to examine the influence that active learning has on the amount of time saved during the training process. Both the performance and the amount of time spent training are analyzed and compared for two different scenarios: training without active learning improvement and training with active learning improvement.

During the tests in which active learning improvement was not used, we trained the deep learning model by making use of a sizable labelled dataset and using the conventional procedures for supervised learning. These trials consistently reached a high level of accuracy, which is indicative of the usefulness of the deep learning model for the categorization of DNA mutations. The extensive and exhaustive training for the complete labeled dataset increased the time needed to train these models.

Modencorporating active learning was our approach to improving inadequate training timeframe. Through active learning, we could effectively select a subset of unlabeled DNA mutations which, in practice, would provide model-appropriate challenge scenarios: instances that were highly informative yet also quite difficult. Subsequently, those mutations were tagged and added to the training set. The model could learn from the most relevant yet uncertain cases. This is captured in both Table 1 and Table 2, which detail results for ten-time execution runs across both scenarios.

Results from our investigation indicate that active learning significantly reduces the time spent on training without sacrificing accuracy in categorization. Deep learning's active learning methodology accelerated convergence of the model compared to the more traditional supervised learning approach, which required a greater number of training epochs. This was achieved through selective labelling of pertinent instances facilitated by active learning techniques. This decrease in training time is noteworthy, particularly in circumstances in which getting labelled data is difficult or expensive.

Table 1: Training time using deep learning.

	Evaluation				Number of Epoch (Early Stopping)	Training Time (second)
	Accuracy	Precision	Recall	F1-score		
1	99.0%	99.0%	99.0%	99.0%	23	3178.702792
2	99.1%	99.1%	99.1%	99.1%	20	3303.412284
3	99.1%	99.1%	99.1%	99.1%	20	3720.917172
4	99.2%	99.2%	99.2%	99.2%	25	6201.880178
5	99.3%	99.3%	99.3%	99.3%	31	7518.152268
6	99.3%	99.3%	99.3%	99.3%	33	5834.808438
7	99.5%	99.5%	99.5%	99.5%	28	5926.236909
8	99.4%	99.4%	99.4%	99.4%	33	4374.317897
9	100%	100%	100%	100%	27	3623.44619
10	100%	100%	100%	100%	37	4914.903106
Average						4859.678

Table 2: Training time using Deep-Active learning model.

Evaluation				Training Time			
Accuracy	Precision	Recall	F1-score	First Training Time (second)	Predict Time (second)	Second Training Time (second)	Total Time with Predict (second)
99.5%	99.5%	99.5%	99.5%	808.1259315	29.30705452	391.6801417	1229.113128
99.52%	99.52%	99.52%	99.52%	817.396971	30.60272074	386.7978294	1234.797521
100%	100%	100%	100%	841.1775959	28.94804478	429.3741367	1299.499777
99.5%	99.5%	99.5%	99.5%	851.2294636	30.49943447	434.3342633	1316.063161
100%	100%	100%	100%	846.6954155	30.04458833	477.5759137	1354.315917
99.7%	99.7%	99.7%	99.7%	912.3629904	30.41996789	447.6847072	1390.467665
100%	100%	100%	100%	985.2940087	30.63900638	513.3931017	1529.326117
99.7%	99.7%	99.7%	99.7%	1081.647869	34.21525335	462.2605217	1578.123644
100%	100%	100%	100%	1135.157587	31.93032336	502.4230978	1669.511008
100%	100%	100%	100%	1509.434553	40.7174468	592.4776373	2142.629637
Average		978.8522		31.73238	463.8001	1474.385	

7 COMPARATIVE ANALYSIS

The methodologies that were examined in earlier studies that were included in Section 2 of this study were compared with the suggested approach. The comparison can be summarized as follows:

An approach of optimum batch bucketing and parallelization of data over several graphic processing units is proposed by the researchers in references [9], [10]. It's possible that the amount and features of the incoming data will have an influence on how successful this method is. The advantages of parallelization and batch bucketing could be restricted if the dataset is very small or if the input sequences have lengths that are generally consistent with one another. In situations like these, the time and effort required to manage simultaneous execution, and the transport of data could be outweighed by the possible benefits, while our suggested strategy provides more flexibility and efficiency.

In the references [14], [18] the Ping-Pong strategy for distributed acceleration of the training of deep neural networks provides various benefits, but there are also some drawbacks to take into consideration, such as an increase in the amount of communication overhead. The Ping-Pong method requires consistent These forms of communication add extra requirements on latency and bandwidth which could impact the speed and efficiency of training. These sorts of communications tend to slow things down, particularly when working with large-scale models or datasets. Our method eliminates the backlog caused by excess communications, so it reduces the total time required for training. As outlined in section [16], the DropSample methodology selectively removes samples to reduce the time needed to train the neural networks. This method is not without its problems, however. It is possible that by omission certain samples from the training dataset, information that could have proved useful is discarded which a loss would occur. These omitted samples might contain essential variations, infrequent occurrences, or beneficial patterns that aid in generalization assailable the entire model and if such samples are tossed aside, it prevents learning important parts of data and most likely hurt overall performance.

This remains true when considering our approach where we use the full training dataset. We take care to ensure that no information considered valuable is discarded. Doing so enables the model to learn from a wide range of patterns, including data variations and unusual examples that can enhance generalization as well as overall performance of models trained on real-world data.

8 CONCLUSIONS

Deep learning is a topic with great potential as it has transformed entire industries by achieving astonishing levels of success across many tasks. However, deep neural networks models usually require a considerable amount of time to be trained, especially when dealing with large-scale datasets. This is more prevalent in the case of DNA mutation classification because analyzing large genomic data sets is crucial for determining disease pathways and designing precision therapeutics. In this regard, this research demonstrates the importance of employing active learning along with LSTM (long short-term memory) networks for deep neural network based DNA mutation classification. The proposed design

which trained and tested the approach using the Cancer Cell Lines Encyclopedia (CCLE) dataset showed that it was possible to overcome the problem of spending lots of time training deep networks on large datasets.

The training process becomes more streamlined when proactive learning techniques are applied. Active learning works best where only samples containing relevant information are chosen for annotation. The efficiency of the model now adds a significant amount value because LSTMs augmenting the model allows it to capture long-term dependencies and trends within DNA mutations more effectively. The suggested approach outperforms the accuracy and time metrics of training with LSTM and deep neural networks without active learning. Comprehensive testing and evaluation have been done. These findings highlight the opportunity provided by active learning to accelerate the training phase of deep network models, especially concerning DNA mutation classification. This gives some perspective towards the other potential uses of active learning in reducing the time required to train deep neural networks. It reveals fundamental aspects towards designing efficient models of deep learning for classifying DNA mutations. All in all, combining active learning with LSTM is a promising outlook for advancement on this topic. Together, they provide solutions to the problem of lengthy training periods for deep neural networks when dealing with large datasets while maintaining a high level of precision While the suggested approach showed significant gains in training efficiency and computational cost, various future possibilities look intriguing. One approach is to use Transformer-based designs (e.g., BERT), which have shown considerable promise in bioinformatics by efficiently capturing long-range genomic connections beyond the limits of LSTM. Another approach is to expand the model to include multi-omics integration, as DNA mutations frequently interact with other biological signals such as RNA expression, epigenetic changes, and proteomics. Using multi-modal learning across these many data sources may give deeper and more complete insights into complicated biological processes.

REFERENCES

[1] H. Sarker, "Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions," SN Computer Science, vol. 2, no. 6, p. 420, 2021.

[2] K. Korfmann, O. E. Gaggiotti, and M. Fumagalli, "Deep learning in population genetics," Genome Biology and Evolution, vol. 15, no. 2, p. evad008, 2023.

[3] W. S. Alharbi and M. Rashid, "A review of deep learning applications in human genomics using next-generation sequencing data," Human Genomics, vol. 16, no. 1, pp. 1-20, 2022.

[4] F. Alharbi and A. Vakanski, "Machine learning methods for cancer classification using gene expression data: A review," Bioengineering, vol. 10, no. 2, p. 173, 2023.

[5] M. Taha and S. Alsaidi, "A review of classification techniques and computer-aided methods used for breast cancer detection."

[6] M. A. Taha, S. A. A. A. Alsaidi, and R. A. Hussein, "Machine learning techniques for predicting heart diseases," in 2022 International Symposium on Innovative Informatics of Biskra (ISNIB), 2022, pp. 1-6.

[7] H. Bi, M. Perello-Nieto, R. Santos-Rodriguez, P. Flach, and I. Craddock, "An active semi-supervised deep learning model for human activity recognition," Journal of Ambient Intelligence and Humanized Computing, pp. 1-17, 2022.

[8] V. Khomenko, O. Shyshkov, O. Radyvonenko, and K. Bokhan, "Accelerating recurrent neural network training using sequence bucketing and multi-GPU data parallelization," in 2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP), IEEE, 2016, pp. 100-103.

[9] M. Wu, C. Li, and Z. Yao, "Deep active learning for computer vision tasks: Methodologies, applications, and challenges," Applied Sciences, vol. 12, no. 16, p. 8103, 2022.

[10] X. An, M. Zhang, and S. Xu, "An active learning-based approach for screening scholarly articles about the origins of SARS-CoV-2," PLOS ONE, vol. 17, no. 9, p. e0273725, 2022.

[11] G. Cong, G. Domeniconi, J. Shapiro, F. Zhou, and B. Chen, "Accelerating deep neural network training for action recognition on a cluster of GPUs," in 2018 30th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD), IEEE, 2018, pp. 298-305.

[12] K. Seo, S. Dodson, N. M. Harandi, N. Roberson, S. Fels, and I. Roll, "Active learning with online video: The impact of learning context on engagement," Computers & Education, vol. 165, p. 104132, 2021.

[13] L. Huang, S. Matwin, E. J. de Carvalho, and R. Minghim, "Active learning with visualization for text data," in Proceedings of the 2017 ACM Workshop on Exploratory Search and Interactive Data Analytics, 2017, pp. 69-74.

[14] S. Sapna, N. S. Sreenivasalu, and K. Paul, "DAPP: Accelerating training of DNN," in 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), IEEE, 2018, pp. 867-872.

[15] C. Limberg, K. Krieger, H. Wersing, and H. Ritter, "Active learning for image recognition using a visualization-based user interface," in Artificial Neural Networks and Machine Learning–ICANN 2019: Deep Learning, Springer, 2019, pp. 495-506.

[16] N. Yang, H. Tang, J. Yue, X. Yang, and Z. Xu, "Accelerating the training process of convolutional neural networks for classification by dropping training samples out," in IGARSS 2018 IEEE International Geoscience and Remote Sensing Symposium, IEEE, 2018, pp. 7141-7144.

[17] P. Kumar and A. Gupta, "Active learning query strategies for classification, regression, and clustering: A survey," Journal of Computer Science and Technology, vol. 35, pp. 913-945, 2020.

[18] C. C. Yang and G. Cong, "Accelerating data loading in deep neural network training," in 2019 IEEE 26th International Conference on High Performance Computing, Data, and Analytics (HiPC), IEEE, 2019, pp. 235-245.

[19] V. Konkol'ová and J. Paralič, "Active learning in data science education," in 2018 16th International Conference on Emerging eLearning Technologies and Applications (ICETA), IEEE, 2018, pp. 285-290.

[20] S. Kee, E. Del Castillo, and G. Runger, "Query-by-committee improvement with diversity and density in batch active learning," Information Sciences, vol. 454, pp. 401-418, 2018.

[21] A. Borodzhieva, I. Tsvetkova, S. Zaharieva, D. Dimitrov, and V. Mutkov, "Active learning strategies applied for teaching the topic of registers," in 2021 12th National Conference with International Participation (ELECTRONICA), IEEE, 2021, pp. 1-4.

[22] S. Schmidt, Q. Rao, J. Tatsch, and A. Knoll, "Advanced active learning strategies for object detection," in 2020 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2020, pp. 871-876.

[23] A. Hekimoglu, M. Schmidt, A. Marcos-Ramiro, and G. Rigoll, "Efficient active learning strategies for monocular 3D object detection," in 2022 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2022, pp. 295-302.

[24] S. Mohamadi and H. Amindavar, "Deep Bayesian active learning: A brief survey on recent advances," arXiv preprint arXiv:2012.08044, 2020.

[25] S. Kee, E. del Castillo, and G. Runger, "Query-by-committee improvement with diversity and density in batch active learning," Information Sciences, vol. 454-455, pp. 401-418, 2018, [Online]. Available: https://doi.org/10.1016/j.ins.2018.05.014.