Real-Time Evolving Deep Learning Models for Predicting Hydropower Generation*

William Girard
Computer and Information Science Department
University of Massachusetts Dartmouth
Dartmouth, MA 02747, USA
wgirard@umassd.edu

Haiping Xu
Computer and Information Science Department
University of Massachusetts Dartmouth
Dartmouth, MA 02747, USA
hxu@umassd.edu

Donghui Yan
Mathematics Department
University of Massachusetts Dartmouth
Dartmouth, MA 02747, USA
dyan@umassd.edu

***Abstract*—Deep learning models have shown great promise for predicting hydropower generation. Previous research has focused on energy output prediction or predictive maintenance using traditional artificial neural networks (ANNs). However, these models lack sustainability in the face of changing environmental conditions. The need for dynamic, real-time modeling becomes apparent in rapidly changing environments, where speed and accuracy of execution are critical. In this paper, we present a framework for real-time evolving deep learning (RT-EDL) models designed to accurately predict hydropower generation on a daily, weekly, and monthly basis. Our evolving model employs backpropagation techniques and a stochastic gradient descent optimizer to continuously fine-tune the model using newly acquired data points in real time. To validate our approach, we conduct a case study using the RT-EDL model and show how the hyperparameters in the evolving model can be adjusted to achieve optimal operation. Our experimental results not only demonstrate the feasibility and effectiveness of our real-time evolving model, but also highlight its superiority over traditional deep learning methods.**

Keywords—deep learning model, real-time evolving model, changing environments, fine-tuning, hydropower generation

I. Introduction

A recent United Nations (UN) report emphasizes the urgent need to ensure a sustainable and habitable future for all, as the window of opportunity is rapidly closing [1]. It is estimated that renewable energy sources will have to double to 60% of global electricity by 2030 to curb the threat of climate change. There is no doubt that climate change has become the overarching issue of the 21st century. A prominent example is the phenomenon of global warming - the continuous increase in the Earth's surface temperature observed since the pre-industrial era (1850-1900). This warming phenomenon is largely attributable to human activities, particularly the burning of fossil fuels, which has led to an increase in the level of "heat-trapping" greenhouse gas in the Earth's atmosphere [2]. Hydropower is one of the most efficient technologies for the production of renewable energy and the largest renewable source of electricity, playing a crucial role in mitigating rising temperatures. However, many existing hydropower plants are in need of modernization. According to the International Renewable Energy Agency (IRENA), the average age of these hydropower plants is close to 40 years, and many countries face significant challenges due to their aging infrastructure [3]. Specifically, hydropower plants in North America and Europe exhibit considerably older ages, averaging around 50 years. Therefore, there is a pressing need to incorporate recent advancements in computer technology, such as artificial intelligence and machine learning (AI/ML), to revolutionize the field of renewable energy utilization. A critical challenge in advancing renewable energy utilization is the accurate prediction of hydropower generation. While traditional machine learning techniques like artificial neural networks (ANNs) have been successful in predicting energy output and facilitating maintenance [4], their reliance on historical data makes them progressively less reliable in the context of changing environmental dynamics, such as shifts in global temperature and weather variations from season to season and year to year. The upcoming retrofit of hydroelectric plants provides an excellent opportunity to integrate real-time evolving models, a machine learning approach that improves accuracy through real-time data [5][6], into the day-to-day operations of hydropower plants. These real-time evolving models can accurately predict hydropower generation on various time scales, enabling plant managers to optimize decision-making and operational efficiency. In this paper, we introduce a framework for real-time evolving deep learning (RT-EDL) models and demonstrates its superiority over traditional machine learning methods. The proposed RT-EDL model is based on a deep neural network that utilizes backpropagation techniques, refines each neural component using a stochastic gradient descent optimizer, and continuously adapts to real-time data for enhanced accuracy.

In our approach, the RT-EDL model can be used to predict future hydropower generation on a daily, weekly and monthly basis, and can be incrementally trained in real time using newly acquired labeled data points as the actual hydropower generation becomes available. Our main contribution is the use of a real-time evolving model that offers significant advantages over traditional AI/ML methods in the face of changing environmental conditions. Unlike traditional ML models, which are trained using historical data and do not take into account new trends in model changes, real-time evolving models continuously assimilate data over time and are able to capture new patterns in the data as a result of environmental changes. This continuous adaptation enhances prediction accuracy by ensuring the model's alignment with the most current data, resulting in more precise forecasts. These characteristics make a real-time evolving model especially advantageous in dynamic, high-speed environments, as evidenced in this research. In our case study, we demonstrate that the introduced real-time evolving model significantly outperforms traditional machine learning methods in predicting hydropower generation.

II. Related Work

There have been many previous efforts in predicting hydropower generation using traditional machine learning methods. Barzola-Monteses et al. attempted to predict the energy output of an Ecuadorian hydropower plant using ANNs [7]. They developed ANN structures based on

* This material is based on work supported by Office of Naval Research (ONR) Grant, MUST IV, University of Massachusetts Dartmouth.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

multilayer perceptron (MLP), long short-term memory (LSTM) and sequence-to-sequence (seq2seq) LSTM. They showed that the MLP univariate and differentiated model outperformed the other architectures based on analysis in several different scenarios. Velasquez and Flores used traditional machine learning methods for predictive maintenance of hydroelectric plants [8]. They implemented deep leaning models using LSTM for early detection of anomalies through fault classification or time-series behavior hydroelectric plant control variable data. Abdulkadir et al. presented the use of multilayer perceptron neural networks for modeling power generation as a function of reservoir variables for two hydroelectric dams in Nigeria [9]. The results of the correlation coefficients show that these networks can reliably model power generation as a function of reservoir variables for future energy forecasting. While the above methods are promising in predicting hydroelectricity generation through traditional machine learning, they may fall short in capturing new data patterns due to environmental changes. In contrast, our approach improves prediction accuracy by continuously adjusting to real-time data and synchronizing the model with the latest available data points.

Previous work related to real-time learning models is summarized as follows. Song et al. introduced an autonomous incremental computing framework and architecture tailored for deep learning in IoT applications [5]. Their approach provides autonomous data diagnostics for deep learning-based IoT systems, aiming to reduce data transfer while employing incremental and unsupervised training techniques. This strategy addresses the challenges posed by the large amount of dynamic raw IoT data generated in an ever-changing environment. Aragón et al. implemented an incremental approach using LSTM models for continuous load forecasting in energy management systems [9]. Their approach was compared with the popular statistical model ARIMA and the results showed that the LSTM algorithm holds great promise in combining continuous load forecasting with incremental learning. Ford et al., proposed a real-time self-adaptive classifier (RT-SAC) for identifying suspicious bidders in the online auction house such as Ebay [10]. In their approach, RT-SAC is initialized on a historical dataset and then incrementally trained to gradually adapt to new bidding data in real time, thus supporting efficient detection of suspicious bidders in online auctions. While the discussed methodologies are effective in adapting to dynamic environments, they are not specifically designed to predict hydroelectricity generation. Instead, we introduce a novel strategy that employs real-time evolving deep learning models designed specifically for predicting hydropower generation. By using real-time learning models, this approach complements existing incremental methods to provide a pragmatic solution for efficiently and effectively predicting future hydropower generation.

III. Real-Time Evolving Deep Learning Models

A. Pre-training the RT-EDL Model

Pre-training of the RT-EDL model was done using a traditional machine learning approach that utilizes TensorFlow's Keras to build a sequential deep learning model. Keras serves as an application programming interface (API) for deep learning tasks, specifically through its sequential model, which consists of stacked layers, each with an input and output tensor. The model starts with an input layer, followed by a hidden layer initialized using the Dense function, forming a structure of interconnected neurons. The output layer is constructed specifically for regression, and thus contains a single neuron for the application of hydropower generation prediction. The model is configured using mean squared error (MSE) as the loss function and mean absolute error (MAE) as an additional metric. The model is trained using the Keras fit function over a specified number of epochs and batch size, with a standardized 20% validation split. Determining the number of neurons in each hidden layer required systematic testing in power of 2 increments. Similarly, the learning rate needed to be manually checked and adjusted from maximum to minimum values. Various batch sizes were also tested, again in power of 2 increments. Table I shows the selected hyperparameters for pre-training of the RT-EDL models.

TABLE I. Manual Tuning of Hyperparameters

Parameter	*Chosen value*	*Min value*	*Max value*
Activation Function	Leaky ReLu	N/A	N/A
Neurons in Hidden Layer 1	128	4	512
Neurons in Hidden Layer 2	64	4	512
Neurons in Hidden Layer 3	32	4	512
Learning Rate	0.003	0.0001	0.1
Epochs	500	N/A	N/A
Batch size	32	2	128

B. Updating the RT-EDL Model

The flowchart for updating the RT-EDL model is shown in Fig. 1. The pre-trained RT-EDL model is used as the initial model for real-time training. As shown in the figure, the most recent plant operational data are used for data point replacement in the moving training window, which contains a fixed number of data points for real-time training. Once the real-time training process is completed, the updated RT-EDL model is ready to be used for real-time energy prediction.

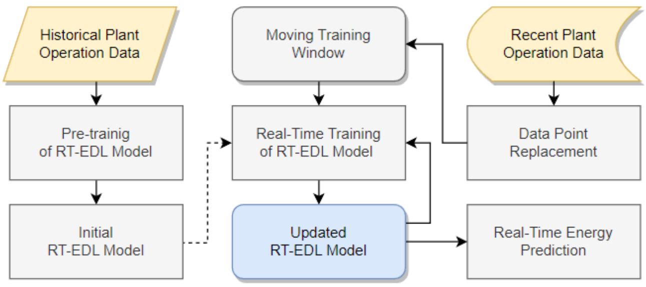

Fig. 1. Flowchart for updating the RT-EDL model

The real-time training process for RT-EDL models described above can be repeated when new data points are acquired. Some or all of the data points in the moving training window can be replaced by new data points. The latest version of the RT-EDL model can then be fine-tuned to support real-time energy prediction using the data points in the moving training window. Finally, the fine-tuned RT-EDL model becomes ready for further real-time training.

C. Real-Time Training

The purpose of real-time training is to fine-tune the RT-EDL model as new data points emerge. The standard deep learning approach involves training the model repeatedly on the entire training dataset for a set number of epochs. The model is then tested using the test dataset to evaluate its performance. This approach is not suitable for real-time implementation of the RT-EDL model. Since the RT-EDL model does not have access to future data points, new training data points must be added to a moving training window over time. The model must then adapt to the new data points in the window, hence the term "evolving model". Before we describe the details of the real-time fine-tuning process, we provide a few definitions for the moving training window.

Definition 3.1 Moving Training Window. A moving training window Ψ is defined as a 2-tuple (*RD*, *ND*), where *RD* is a list of recent data points kept in the window and *ND* is a list of newly acquired data points added to the window.

Definition 3.2 Window Size. The window size of a moving training window Ψ is the number of data points contained in the window that consists of both recent data points (i.e., Ψ.*RD*) and newly acquired data points (i.e., Ψ.*ND*). For a window size of n, n data points will be used to fine-tune the RT-EDL model.

Definition 3.3 Window Speed. The window speed of a moving training window Ψ is the number of new data points to be added to Ψ for each round of training and prediction to fine-tune the RT-EDL model. Let the window size be n and the window speed be m, m new data points and m oldest data points will be added to and removed from Ψ, respectively.

Fig. 2 shows an example of a moving training window with a window size of 6 and a window speed of 3. As shown in the figure, the number of incoming data points is set to 3. Since the window size is 6, when 3 new data points are added to the window, the oldest 3 data points must be removed. It is important to note that the window speed and window size must be chosen appropriately for specific training and prediction purposes. For example, if monthly energy prediction is required, a window size of 6 would be too small and may result in an underfitting of the model. Conversely, if daily energy prediction is required, a window of 30 could be too large and may result in model overfitting.

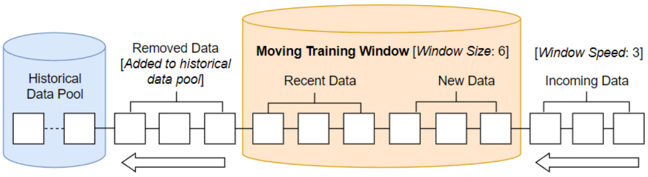

Fig. 2. An example of moving training window

The model is used to predict hydropower generation for the next day, the next week, and the next month outside of the moving training window. To evaluate the accuracy of both the traditional approach and our approach using RT-EDL models, we define a few evaluation metrics as follows.

Definition 3.4 Prediction Accuracy. The prediction accuracy *pa* for a singular energy prediction is defined as in (1), where x is the actual value to be predicted, and y is the predicted value.

$$pa = 1 - \frac{|x-y|}{max\ (|x|,|y|)} \tag{1}$$

In our approach using RT-EDL models, the prediction accuracies on a daily, weekly, and monthly basis are compared with the average prediction accuracy *avgpa* for a number of singular days, which is defined as in (2).

$$avgpa = \sum_{i=1}^{n} pa_i = \sum_{i=1}^{n} \left(1 - \frac{|x_i - y_i|}{\max(|x_i|,|y_i|)}\right) \tag{2}$$

where n is the number of data points in the test dataset, and x_i and y_i are the actual value to be predicted and the predicted value in data point i ($1 \le i \le n$), respectively.

Before the model can be evaluated, it must be fine-tuned using the data points in a newly created moving training window. We refer to the collection of data points in the window as the batch data. Algorithm 1 details the fine-tuning process for the RT-EDL model.

Algorithm 1 Fine-tuning the RT-EDL Model

Input: Current RT-EDL model Φ, current moving training window Σ of size τ, and η new labeled data points
Output: Fine-tuned RT-EDL model Φ'

1. Remove the η oldest labeled data points from Σ.
2. Add η new labeled data points to the updated Σ.
3. Set the maximum epochs to σ
4. Initialize the number of iterations i to 0
5. **while** $i < \sigma$
6.   **for** each labeled data point α in Σ
7.   Train model on data point α using backpropagation
8.   Calculate the training accuracy *ta* as in (1)
9.   Calculate the average training accuracy *avgta* as in (2)
10.   **if** *avgta* does not improve **and** *avgta* ≥ 90 **break**;
11. $i = i + 1$
12. **return** Fine-tuned RT-EDL model Φ'

As seen in Algorithm 1, the new batch data is created by removing η oldest labeled data points from the moving training window Σ and adding η newly labeled data points. The batch data is then used to train RT-EDL for a maximum of σ epochs. The epoch-based stopping condition is used to ensure fast computational efficiency, which is a key requirement of this methodology. Additional stopping conditions that stabilize model performance are intended to reduce overfitting by identifying if the model is no longer improving. Combined with a 90% average training accuracy threshold, our method ensures that the model stops training only when the accuracy is high enough. Note that our methods for calculating the training accuracy *ta* and the average training accuracy *avgta* are consistent with the methods for calculating the prediction accuracy *pa* and the average prediction accuracy *avgpa* defined in (1) and (2), respectively. Finally, the fine-tuned RT-EDL model is returned and can be used in energy prediction applications.

Algorithm 2 shows the complete process of pre-training, real-time training and predicting hydropower generation on a daily, weekly, and monthly basis using the RT-EDL model.

Algorithm 2 Real-Time Training and Prediction (Reactive)

Input: Historical data pool, window size τ, and window speed η
Output: *null*

1. Pre-train RT-EDL model Φ using historical data points.
2. Create a new moving training window Σ of size τ
3. Add the most recent τ data points in historical data pool to Σ
4. Initialize number of days d to 0
5. **while** (true)
6.   **for** i from 1 to η
7.   Predict and print out the next day energy generation
8. $d = d + 1$
9.   **if** number d day is the first day of a week
10.   predict and print out next week's energy generation
11.   **else if** number d day is the first day of a month
            predict and print next month's energy generation
12.   Invoke *Algorithm* 1 with model Φ, window Σ and η newly labeled data points; obtain fine-tuned RT-EDL model Φ'
13.   Let current RT-EDL model Φ be Φ'

This process starts with a pre-training phase using historical data points as described in Section III.A and the creation of an initial moving training window. The RT-EDL model initiates the prediction process using the pre-trained model and provides the predicted next day's energy generation. Subsequently, the model extends its predictions to weekly or monthly intervals, contingent upon whether the current day marks the beginning of a week or a month. Following this, Algorithm 1 is invoked to update the moving training window and fine-tune the RT-EDL model. This process (line 6 through 13 in Algorithm 2) is repeated. Given the reactive nature of the algorithm, the cycle of fine-tuning the model and predicting energy generation will continue until it is manually terminated.

IV. Optimizing Hyperparameters

In this section, we show how to prepare the training and the test datasets, and how to optimize the hyperparameters of the RT-EDL model, including the window size and the window speed, through a series of simulations.

A. *Dataset Preparation*

The first step in obtaining a suitable dataset for energy prediction is to find an existing dataset with energy outputs of hydropower plants. These data points must be suitable for a real-time environment, hence the need for daily energy output. The output parameter, or label, is the daily energy output of the hydropower plant in megawatt-hours (MWh). Table 1 lists the input parameters selected for the test dataset.

TABLE II. Input parameters

Parameter	*Units*	*Short Description*
Day	Unitless	1-365 or 1-366 on leap year
Temperature	Fahrenheit	Average daily temperature
Temperature Departure	Fahrenheit	Temperature departure from historical mean
Heating Degree Days	Unitless	Expected energy used to heat a house for a singular day
Cooling Degree Days	Unitless	Expected energy used to cool a house for a singular day
Precipitation	Inches	Daily recorded rainfall
Stream Flow	Cubic feet/second	Cubic feet per second of the river attached to the dam

Most of the input parameters listed in Table 1 were selected based on the previous work on comprehensive identification of relevant variables affecting hydropower generation [11]. The parameters related to degree days assume that temperature of 65°F requires neither heating nor cooling to be comfortable. If the daily temperature mean (daily high plus daily low divided by 2) exceeds 65°F, the difference between the mean and 65°F is the cooling degree days. Conversely, if the daily temperature mean is below 65°F, the difference between 65°F and the mean is considered heating degree days [12].

We used an energy dataset, called RectifHyd, to obtain monthly hydro generation estimates for approximately 1,500 power plants in the United States. The time span chosen for analysis is six years, from 2015 to 2020. The U.S. Geological Survey (USGS) offers a free service that generates streamflow data at 15-minute intervals, which was then averaged into daily data entries. The National Weather Service offers a service called NOWData that, upon selecting a weather station, generates a table containing daily data entries for a given month. Temperatures, precipitation, temperature deviations, descending temperature days, and ascending temperature days are extracted from this resource. Finally, six years of monthly data for each hydroelectric plant were simulated, resulting in 2,192 daily data points. Subsequently, all necessary data was compiled, resulting in the creation of the following two datasets.

- Dataset 1 (DS1): A specific hydroelectric plant, Black Canyon Dam in Idaho, was selected from the RectifHyd dataset. Flow data from the neighboring river, the Payette River, and meteorological data from the nearest weather station, Plaza, Idaho, were collected and added to DS1.
- Dataset 2 (DS2): A specific hydropower plant, Flaming Gorge Dam in Utah, was selected from the RectifHyd dataset. Flow data for the Green River and meteorological data from the nearest weather station, Dutch John, Utah, were collected and added to DS2.

In the case study presented in Section V, the traditional method is trained using the first three years of data and then tested on the last three years of data. In our real-time approach, the RT-EDL model is pre-trained on three years of data. The remaining three years are used for real-time training and prediction. In constructing deep neural network models, the best optimizer and activation function were found to be RMSprop and Leaky ReLU, respectively. Table III lists the remaining hyperparameters manually selected for DS1 and DS2 based on experiments.

TABLE III. Manual Tuning of Hyperparameters

Parameter	*Chosen Value*	*Min Value*	*Max Value*
Hidden Layer 1 (DS1)	1028	4	1028
Hidden Layer 2 (DS1)	16	4	1028
Hidden Layer 3 (DS1)	4	4	1028
Hidden Layer 1 (DS2)	1028	4	1028
Hidden Layer 2 (DS2)	256	4	1028
Hidden Layer 3 (DS2)	8	4	1028
Learning Rate	0.003	0.0005	0.1
Epochs	35	5	50

To avoid overfitting and ensure that the desired accuracy is achieved, we used 1028 neurons in the initial hidden layer for both datasets. The number of epochs per training batch was limited to 50 to ensure that real-time training could be completed in the expected time. After tuning the regular hyperparameters of the deep neural network models, the window size and window speed had to be optimized. In the following sections, we use DS2 for these experiments.

B. *Optimizing Window Sizes*

We first conducted experiments to find the optimal window size for daily accuracy. Real-time prediction of daily accuracy requires only one new data point, so the window speed can only be 1. Fig. 3 shows the daily accuracy results versus the window size.

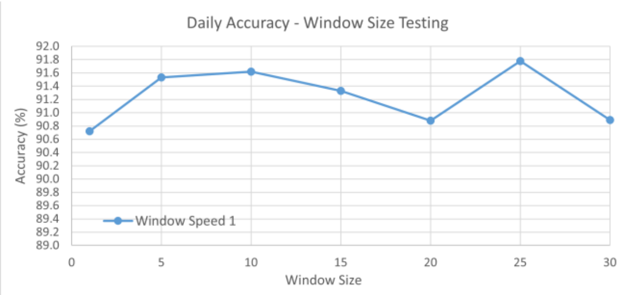

Fig. 3. Daily accuracy – window size testing

As can be seen from the figure, although a window size of 25 has the highest daily accuracy of 91.78%, all the tested window sizes are very close to each other in terms of accuracy. Since the smaller the window size, the more efficient the training, we consider a window size of 10 to be the optimal window size for daily accuracy. Similarly, we tested the window size for weekly predictions. Fig. 4 shows the accuracy results versus window size, with multiple lines in the figure corresponding to different window speeds.

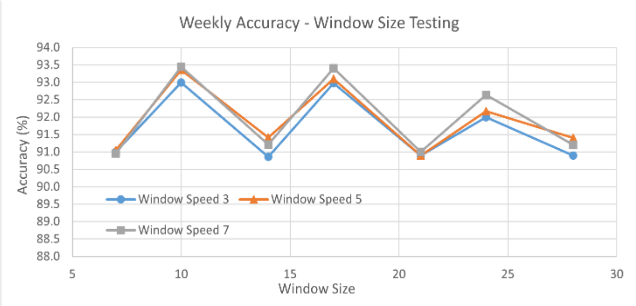

Fig. 4. Weekly accuracy – window size testing

Fig. 4 shows that the weekly accuracies for all window sizes are once again very close. The most prominent window sizes are 10 and 17, both hovering around 93% accuracy. A window size of 10 was chosen for its efficiency. Finally, the window size for monthly prediction was tested and the results are shown in Fig. 5. The figure indicates that the optimal window size is 40. Using this window size, the model not only achieves an accuracy of about 87%, but also the window size is small enough to be efficient.

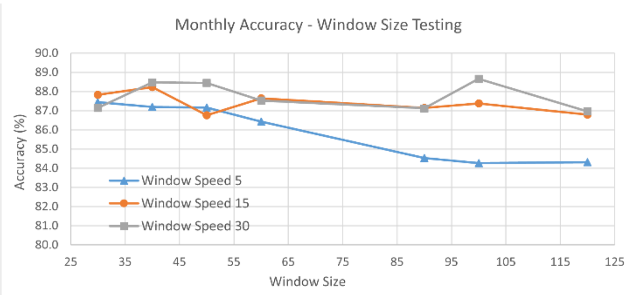

Fig. 5. Monthly accuracy – window size testing

C. *Optimizing Window Speeds*

Now that all optimal window sizes have been selected, optimal window speeds for the monthly and weekly predictions must be found. Fig. 6 shows the window speed tests for the weekly and monthly predictions.

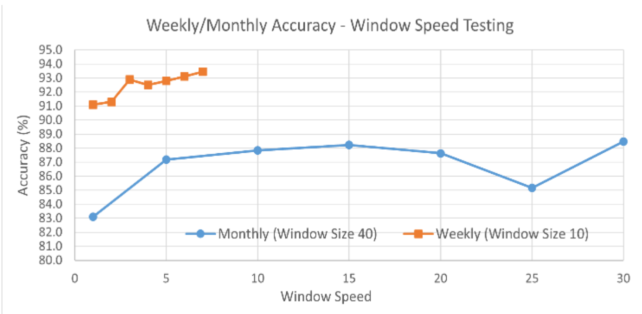

Fig. 6. Weekly/Monthly accuracy – window speed testing

As we can see from Fig. 6, the accuracies using window speeds of 3-7 for weekly predictions are all very close to each other. The window speed of 7 was chosen because it directly corresponds to a week's worth of data and has high accuracy. For monthly prediction, the optimal window speed is 30 due to its high accuracy as well. These experiments show that window speed has a large effect on accuracy, while window size has a small effect. In Section V, these optimal window sizes and window speeds are used for the experiments to compare the accuracies between traditional machine learning methods and the real-time approach using RT-EDL models.

V. Case Study

Using optimized hyperparameters, we demonstrate that our approach is feasible and effective, and outperforms traditional machine learning methods. All experiments were conducted on a workstation equipped with a 2.8GHz Intel Core i7-1165G7 processor and 16 GB of RAM.

A. *Accuracy Analysis of the Traditional Approach*

Initial testing has shown that the traditional method (no real-time training) using TensorFlow is likely to suffer from overfitting. Overfitting is when there are too many neurons, resulting in an overly complex model with reduced accuracy. Manually removing a certain percentage of neurons (called *dropout*) can solve this problem. Table IV shows the optimal percentage of dropout and model accuracy.

TABLE IV. Accuracy using the Tradtional method

Dataset	*Dropout (%)*	*Accuracy (%)*
DS1	30	82.1
DS2	40	85.1

Additionally, useful graphs can be created to demonstrate the variability of the traditional methods. Fig. 7 illustrates the daily testing accuracy of model predictions versus the number of days for the two datasets DS1 and DS2.

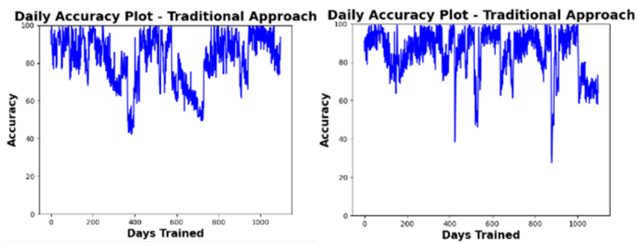

Fig. 7. Daily testing accuracy using traditional approach for DS1 & DS2

The graphs reveal substantial downward spikes in accuracy, with certain predictions dropping to nearly 20%. The shape of these two graphs displays significant variability, characterized by a sudden transition from a long period of low accuracy to high accuracy. This pattern implies that the model encounters challenges in making accurate predictions.

B. *Accuracy Analysis of the RT-EDL Model*

For the real-time approach using the RT-EDL model, dropout experiments were also performed on both datasets. Fig. 8 shows the daily testing accuracy graphs for datasets DS1 and DS2, with fewer downward spikes in daily accuracy. Using the optimal window speed 1 and window size 10, DS1 was found to have no need for dropout. Based on the experimental results, the average daily prediction accuracy was 90.8%. This is an 8.7% improvement over the traditional method. For dataset DS2, using a 30% dropout rate, window speed 1 and window size 10, the average daily prediction accuracy increased to 91.6%. This is a 6.5% improvement over the traditional method.

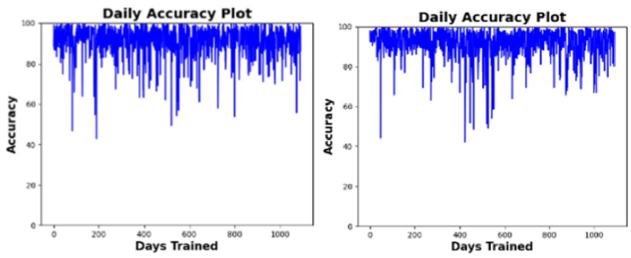

Fig. 8. Daily testing accuracy graphs – datasets DS1 & DS2

Fig. 9 shows the weekly testing accuracy graphs for datasets DS1 and DS2 with window speed 7 and window size 10. The average weekly prediction accuracy for dataset DS1 is 92.4% with no dropout. The average weekly prediction accuracy for dataset DS2 is 93.4% with a 30% dropout rate. Compared to the traditional method, there is an improvement of 10.3% and 8.3%, respectively.

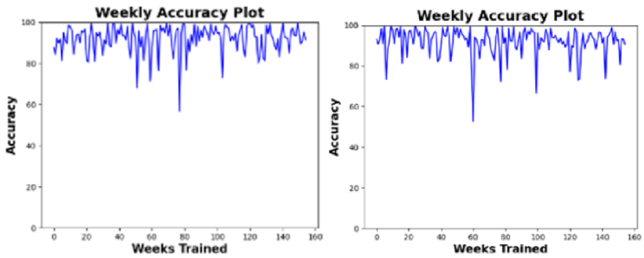

Fig. 9. Weekly testing accuracy graphs – datasets DS1 & DS2

Fig. 10 shows the monthly testing accuracy graphs for datasets DS1 and DS2 with window speed 30 and window size 40. The average monthly prediction accuracy for dataset DS1 is 88.9% with no dropout. The average weekly prediction accuracy for dataset DS2 is 88.4% with 30% dropout. Compared to the traditional method, DS1 and DS2 show an improvement of 6.8% and 3.3%, respectively.

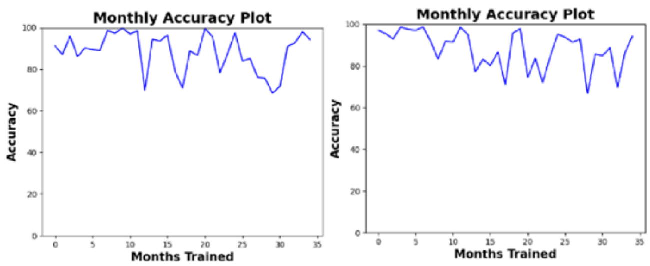

Fig. 10. Monthly tetsing accuracy graphs – datasets DS1 & DS2

It is evident that there are occasional downward spikes in accuracy using RT-EDL models. However, these downward spikes are infrequent, especially compared with the graphs in Fig. 7. It is worth noting that such downward spikes are more frequent in the daily graphs because of the greater impact of outliers on accuracy and challenging data points, such as those influenced by extreme weather (e.g., hurricanes, heavy rainfall, and floods). In contrast, the weekly and monthly graphs show almost zero downward spikes. The reduction in downward spikes can be attributed to the higher window speeds, i.e., 7 for the weekly and 30 for the monthly graphs. The increased number of newly acquired data points mitigates the impact of outliers and makes the predictions more resilient to extreme outliers.

VI. Conclusion And Future Research

Renewable energy research has never been more relevant as climate change continues to dominate scientific discussion in the 21st century. However, the traditional machine learning approach was not suitable for real-time implementation and the accuracy was rather average. In this paper, we introduced the RT-EDL model, which is pre-trained using a historical dataset and can evolve over time. We discussed in detail the tuning of hyperparameters including the window speed and the window size. The experimental results show that our RT-EDL-based approach outperforms the traditional machine learning method.

In future studies, we plan to investigate correlations between input parameters to obtain accurate feature importance rankings and to investigate whether model predictions follow typical climate changes such as global warming trends. To improve the efficiency and effectiveness of our approach, we will further study automated methods to determine the optimal number of neurons in the hidden layers, the appropriate maximum number of epochs, and other suitable stopping conditions. Finally, we will consider developing an RT-EDL model that not only changes the weights, but also automatically adapts to environmental changes by self-correcting the neural network structure.

References

[1] United Nations, "UN chief calls for renewable rnergy 'revolution' for a brighter global future," *UN News*, Global perspective Human stories, United Nations, January 14, 2023. Retrived on September 1, 2023 from https://news.un.org/en/story/2023/01/1132452

[2] NASA, "What Is Climate Change?" *Newsletter*, Earth Science Communications Team, NASA, 2018. Retrieved on November 1, 2023 from https://climate.nasa.gov/what-is-climate-change/

[3] IRENA, "The changing role of hydropower: challenges and opportunities," *IRENA Newsletter*, International Renewable Energy Agency (IRENA), February 2023. Retrieved on March 1, 2023 from https://www.irena.org/Publications/2023/Feb/The-changing-role-of-hydropower-Challenges-and-opportunities

[4] J. Bernardes, M. Santos, T. Abreu, L. Prado, D. Miranda, R. Julio, P. Viana, M. Fonseca, E. Bortoni, and G. S. Bastos, "Hydropower operation optimization using machine learning: a systematic review," *AI*, vol. 3, no. 1, 2022, pp. 78-99.

[5] M. Song, K. Zhong, J. Zhang, Y. Hu, D. Liu, W. Zhang, J. Wang, and T. Li, "In-situ AI: towards autonomous and incremental deep learning for IOT systems," In *Proceedings of the 2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Vienna, Austria, 2018, pp. 92-103.

[6] B. J. Ford, H. Xu, and I. Valova, "A real-time self-adaptive classifier for identifying suspicious bidders in online auctions," *The Computer Journal*, vol. 56, no. 5, 2012, pp. 646-663.

[7] J. Barzola-Monteses, J. Gómez-Romero, M. Espinoza-Andaluz and W. Fajardo, "Hydropower production prediction using artificial neural networks: an Ecuadorian application case," *Neural Computing and Applications*, vol. 34, no. 16, 2022, pp. 13253-13266.

[8] V. Velasquez and W. Flores, "Machine Learning Approach for Predictive Maintenance in Hydroelectric Power Plants," In *Proceedings of the 2022 IEEE Biennial Congress of Argentina (ARGENCON)*, San Juan, Argentina, 2022, pp. 1-6.

[9] G. Aragón, H. Puri, A. Grass, S. Chala and C. Beecks, "Incremental deep-learning for continuous load prediction in energy management systems," In *Proceedings of the 2019 IEEE Milan PowerTech*, Milan, Italy, June 23-27, 2019, pp. 1-6.

[10] T. S. Abdulkadir, A. W. Salami, A. R. Anwar, and A. G. Kareem, "Modeling of hydropower reservoir variables for energy generation: neural network approach," *Ethiopian Journal of Environmental Studies and Management*, vol. 6, no. 3. 2013, pp. 310-316.

[11] F. Zhou, L. Li, K. Zhang, G. Trajcevski, F. Yao, Y. Huang, T. Zhong, J. Wang and Q. Liu, "Forecasting the evolution of Hydropower Generation," In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*, Virtual Event, CA, USA, July 6-10, 2020, pp. 2861-2870.

[12] US Department of Commerce, "What are heating and cooling degree days," *National Weather Service*, National Oceanic and Atmospheric Administration, Key West. FL, USA. Retrived on November 1, 2023 from https://www.weather.gov/key/climate_heat_cool.