Forecasting River Turbidity using Innovative Machine Learning Techniques

Shaurya Swami, Graduate Research Assistant, Department of Computer Science, University of Vermont, Burlington, VT, Shaurya.Swami@uvm.edu

Kristen L. Underwood, Research Assistant Professor, Department of Civil & Environmental Engineering, University of Vermont, Burlington, VT, Kristen.Underwood@uvm.edu

Safwan Wshah, Assistant Professor, Department of Computer Science, University of Vermont, Burlington, VT, Safwan.Wshah@uvm.edu

Dany Davis, Geologist/Project Manager, Stream Management Program, NYC Department of Environmental Protection, Queens, NY, davisd@dep.nyc.gov

Donna M. Rizzo, Professor, Department of Civil & Environmental Engineering, University of Vermont, Burlington, VT, Donna.Rizzo@uvm.edu

Abstract

Turbidity is a vital metric of water quality that has adverse effects on aquatic life. Turbidity can promote pathogen growth, carry harmful contaminants, and severely impact the taste and odor of drinking water. Up to 40% of New York City's (NYC) unfiltered drinking water supply is from the Ashokan Reservoir in the southeastern Catskill Mountains, which is prone to excess turbidity levels originating from stream erosion into glacial legacy sediment. The U.S. Geological Survey has 29 high-frequency turbidity monitoring stations on 12 streams in the 497 km^2 catchment tracking the spatial and temporal production of turbid streamflow. To manage drinking water operations more effectively, the NYC Department of Environmental Protection would benefit from up to a seven-day prediction of river turbidity levels. Currently, traditional regression models face challenges in producing such estimates. Newer computational tools, from the rapidly growing field of Machine Learning (ML), hold great potential for forecasting daily turbidity data. For example, Recurrent Neural Networks (RNNs) have proved valuable in learning patterns in daily streamflow time series for prediction. A sub-category of RNNs, Long Short-Term Memory (LSTM) models, have shown competence in forecasting turbidity. Another type of ML model, called a Gated Recursive Unit (GRU), suggests faster computational speeds over LSTMs in various domains; however, GRUs have not been commonly applied to hydrological data series to date. We aimed to explore the application of ML algorithms to predict turbidity for the Stony Clove watershed in the Ashokan Reservoir catchment and compare the performance of RNN, GRU, and LSTM to one another. We leveraged time series data from several strategically distributed sensor stations for our analysis; furthermore, we employed discharge and meteorological inputs (e.g., precipitation, soil moisture, soil temperature from NY Mesonet) as input features to forecast daily turbidity. We hypothesized that LSTMs would perform better than GRUs and RNNs for forecasting turbidity. Our results found that LSTMs had the best overall performance (all 3-error metrics) for every sensor. RMSE values ranged from 5-18 for the algorithms with only 1 sensor having a slightly lower GRU RMSE value compared to the LSTM. Overall, we observed that LSTMs had the best performance while GRUs had the fastest computation time.

Keywords

RNN, LSTM, GRU, Time Series, Turbidity prediction, Turbidity, Suspended Sediment, Discharge, Watershed management, Reservoir management

Introduction

Turbidity in surface water is a strong indicator of the presence of high levels of bacteria, pathogens, and particles, which may shield harmful organisms from disinfection processes. Thus, consuming highly turbid water containing harmful bacteria may cause minor to severe health-related issues such as nausea, cramps, headaches, and more (Mukundan et al., 2013). Aside from being an indicator of potential health impacts, turbidity affects the taste and odor of the water itself and can impact proper functioning of household appliances or industrial operations when affected surface water is used as a potable water source (Iglesias et al., 2014). Operators of drinking water reservoirs would therefore benefit from forecasts of turbidity to manage reservoir operations more optimally.

The Catskill and Delaware river systems form one of the largest unfiltered surface water supply systems in the world supplying potable water to New York City (Wang et al., 2021). Up to 40% of New York City's (NYC) unfiltered drinking water supply is from the Ashokan Reservoir in the southeastern Catskill Mountains, which is prone to excess turbidity levels (Mukundan et al 2013; McHale and Siemion, 2014). Suspended sediment has been attributed as a major source of turbidity that ends up in the reservoir, originating from stream erosion into glacial legacy sediment (Mukundan et al., 2013; Wang et al., 2021). The U.S. Geological Survey has 29 high-frequency turbidity monitoring stations on 12 streams in the 497 km^2 catchment tracking the spatial and temporal production of turbid streamflow. To manage drinking water operations more effectively, the NYC Department of Environmental Protection would benefit from up to a seven-day prediction of river turbidity levels for optimal management of this drinking water reservoir. Turbidity is strongly correlated to discharge, which itself is correlated to meteorological variables such as precipitation amount, intensity, as well as catchment attributes such as soil moisture and temperature. These correlations suggest the potential to predict turbidity into the future to support reservoir operations.

Traditional turbidity prediction methods, such as regression models, have been shown to outperform other methods in terms of accuracy (Wang et al., 2021; Meyers et al., 2017). However, these methods can be limited by the complex nature of river systems, including memory effects and feedbacks (Iglesias et al., 2014), and may not capture the nonlinear dynamics of the system. A recent study demonstrated that time-variant models outperformed static regression models in terms of log-NSE and mean bias across lead times (Wang et al., 2021), while catchment management practices can significantly influence the relationship between turbidity and discharge, leading to significant scatter about a linear relationship (Iglesias et al., 2014). Other traditional methods include probabilistic forecasting and advanced prediction using machine learning techniques such as Hammerstein-Wiener and neural networks (Gaya et al., 2017). However, these methods can be limited by their difficulty in capturing complex relationships and high variability of turbidity levels (Shi et al., 2022). On the other hand, Recurrent Neural Network (RNN) models, such as Gated Recurrent Units (GRUs) and Long Short-Term Memory (LSTM) models, have shown better performance in time series forecasting and have the potential to improve turbidity prediction in reservoir management (Shi et al., 2022).

Our objective is to build predictive models that will allow the NYC water management group to forecast future (ideally a week or longer) turbidity levels based on limited, and easily available, discharge and meteorological input variables. We chose to build and compare the performance of an RNN, GRU, and LSTM on 5 out of the 6 primary sensors in the Stony Clove sub-basin (6^{th} sensor lacked various variables). Overall, the purpose of this research is to predict turbidity ahead to support reservoir management.

Methods

Study Area and Data

The upper Esopus Creek drains to the Ashokan Reservoir located in the Catskill Mountains of New York. The mountainous stream is a high energy system prone to episodes of elevated turbidity originating from stream erosion into glacial legacy sediment during flooding events (Mukundan et al., 2013; Wang et al., 2021). This work focuses on the Stony Clove tributary catchment (84.2 km^2) of the Upper Esopus Creek watershed (Figure 1), which connects to the Ashokan Reservoir (the second largest reservoir by volume, 466 million m^3) via the Esopus Creek. The Stony Clove sub-basin is monitored by 6 primary sensors and 14 secondary data sensors measuring discharge and various water quality parameters. We focus on this Stony Clove study area initially to develop and evaluate performance of predictive models in this most densely-sensored sub catchment, with a future plan to transfer our learnings to the larger Esopus Creek watershed and inflows to the Ashokan Reservoir.

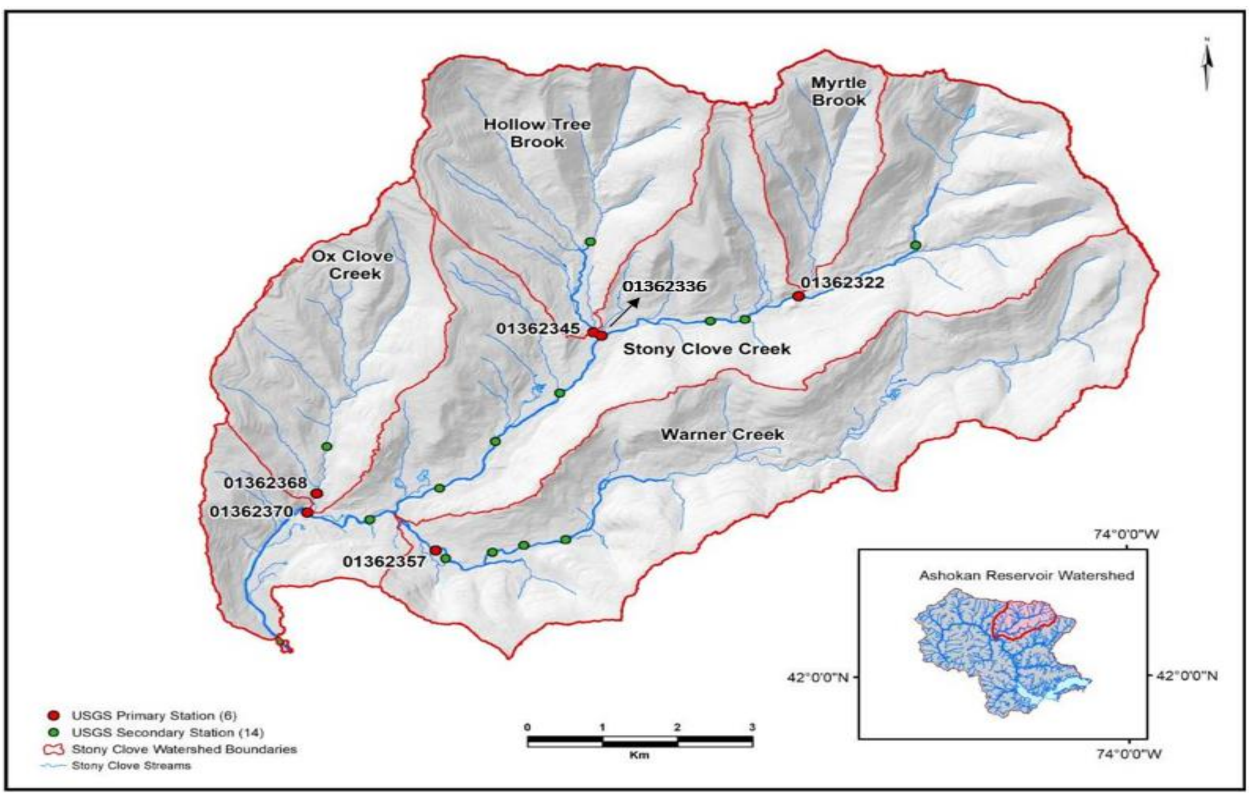

Figure 1. Displays a map of the Stony Clove catchment along with labeled primary sensors (red dots) and secondary sensors (green dots). Source – Created by Dany Davis

Table 1. Offers insight into USGS sensors and the type of parameters they observe.

Gage Number (sensor)	Stream	Gage Elevation (ft Above NAVD)	Upstream Drainage Area (mi ²)
01362370	Stony Clove Creek	947.52	30.9
01362368	Ox Clove Creek	1020	3.83
01362357	Warner Creek	1119.95	8.71
01362336	Stony Clove Creek	1280	9.25
01362322	Stony Clove Creek	1530	1.81

This study utilized data from two sources: 1) USGS and 2) NY Mesonet (Brotzge et al., 2021) Daily time interval USGS open-source data were downloaded from their website for each of the sensors listed in Table 1. Similarly, meteorological data (e.g., precipitation, soil moisture, soil temperature) were retrieved from the NY Mesonet website via a data download request. Since the frequency of the USGS data were observed in daily increments, the Mesonet data were cleaned and down sampled from 5-minute increments to daily mean frequencies to match the USGS timestamps. The widely used interpolation method in python's pandas library was used to impute all missing data (Lepot et al., 2017). Since outliers were rare (< 8 – 10 in the entire dataset) , they were not removed in order to analyze occasional high peaks to uncover plausible trends. We examined correlation coefficients (R) for the feature variables associated with our target variable (Turbidity) and selected the 6 feature variables (Turbidity, Discharge, Suspended Sediment Concentration (SSC), Sediment Suspended Load (SSL), Precipitation local, and Precipitation incremental) with values of R > 0.5 as our input variables; we then plotted these 6 timeseries to gain visual insight into correlating trends. We trained 3 individual models (RNN, GRU, and LSTM) on data from each of the 5 primary sites (Table 1). Each model was trained on data solely from 1 site/sensor; thus, we have 3 models for each sensor.

Algorithms

To forecast turbidity, we compared three algorithms in terms of increasing structural complexity: namely RNN, GRU, and LSTM (Figure 2). RNNs, which have a simple architecture compared to GRU and LSTM, also have limited memory retention and are not ideal for modeling long-term dependencies (Hussain et al., 2018). GRU and LSTM are more complex variants of RNNs that include gating mechanisms. GRUs have two gates, a reset gate that controls how much of the past information to forget, and an update gate that decides how much of the new information to retain. LSTMs have three gates, an input gate that controls how much of the new information to store, a forget gate that controls how much of the past information to forget, and an output gate that controls how much of the cell's memory to reveal as output.

An RNN is capable of modeling a collection of sequential data by recognizing and learning patterns based on previously seen iterations of data. They work particularly well with time series data since they maintain internal memory that re-circulates activations along with input data as the model iterates through the dataset. Yet, RNNs have limited memory retention and do not model long-term dependencies well because of exploding or vanishing gradients (Hussain et al., 2018). Essentially, RNNs can only remember information for a short time with some only looking at the sliding window length to make future predictions. GRU and LSTM were created to address the shortcomings of RNNs; therefore, we decided to compare the performance of all three algorithms. Not only do GRU and LSTM models alleviate the short-term memory issues (alleviate vanishing gradients) of their predecessor, but they also provide a wider array of learnable weights/parameters (e.g., learning rates, input bias, output bias) to optimize model performance (Shi et al., 2022). Additionally, the improved physical architecture of GRU and LSTM models allows better control of the input flows, and therefore, more control over the outputs. In terms of model hierarchy, GRUs have a less complex model architecture compared to LSTMs that enable them to perform computationally faster (Sadon et al., 2021). Yet the more intricate infrastructure of the LSTMs theoretically makes them perform better on larger datasets compared to GRUs (Pirani et al., 2022).

The bottom half of Figure 2 shows the concept of a sliding window to highlight the length of prior time steps used to predict a specified number of future time steps. Due to time constraints on training models, our algorithms currently take in 4 days of daily data and forecast turbidity values for one day into the future. Our initial attempts to train models using 15-minute data were time consuming, so we opted to use daily data instead to ensure timely results. We acknowledge that this approach may not be suitable for real-time applications that require more frequent updates, but it allowed us to compare the performance of the algorithms using consistent training data and hyperparameters. In future work, we plan to explore the use of more granular data and real-time applications to address this limitation. All models were designed using ML libraries like Scikit-learn and TensorFlow. We used three error metrics to compare the model performance: Mean Absolute Error (MAE), Mean Absolute Percent Error (MAPE), and Root Mean Squared Error (RMSE). To facilitate comparison across algorithms, we used consistent hyperparameters to train all three models – RNN, GRU, and LSTM (Table 2).

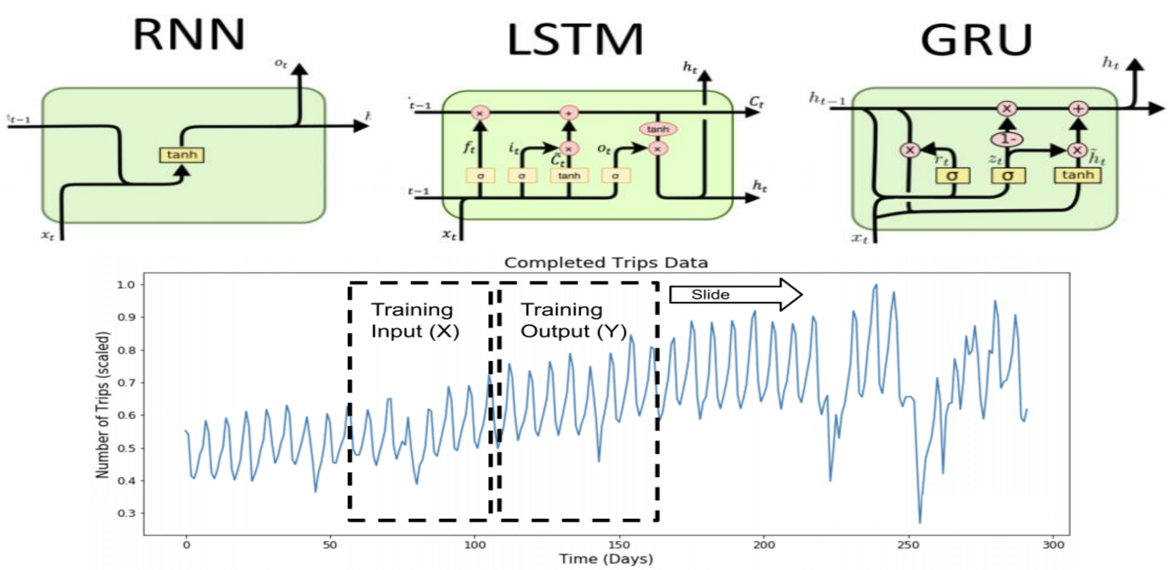

Figure 2. Shows algorithm structural complexity (RNN being simplest, GRU in the middle, and LSTM being the most complex). Bottom schematic spotlights the concept of sliding window in time series. Source – 1) http://dprogrammer.org/rnn-lstm-gru, 2) https://machinelearningmastery.com/lstm-model-architecture-for-rare-event-time-series-forecasting/

Table 2. Represents final input features, their observation frequency, and their data source.
Note - Algorithms were all trained on the same parameters.

Batch Size	Window Length	Epochs	Learning Rate	Activation	Loss Function	Optimizer	Layers	Units	Train/Test split
8	4	50	0.5	LeakyReLU	MAE	Adam	3 Algo + 1 Dense	64	80/20

Results & Discussion

Our 6 selected input features for each algorithm are summarized below (Table 3).

Table 3. Represents final input features, their observation frequency, and their data source.
Note – Suspended Sediment Concentration (SSC) and Suspended Sediment Load (SSL)

Feature	Observation Frequency	Data Source
Discharge	Daily	USGS
Turbidity	Daily	USGS
Estimated SSC	Daily	USGS
Computed SSL	Daily	USGS
Precipitation Local	Daily	NY Mesonet
Precipitation Incremental	Daily	NY Mesonet

LSTMs mostly dominated the best performance among the three algorithms (Table 4), while the GRUs proved to be the computationally fastest. According to Chung et al. (2014), GRUs are typically faster than LSTMs due to having a smaller number of parameters and internal states, which leads to reduced computational requirements during training and testing. LSTMs tended to have the lowest error values (best fit between forecasted values and actual values) across all 3 performance metrics columns for each of the 5 sensors. In two cases, the GRU seems to have a slightly lower error value compared to the LSTM but that only happens in one performance metrics column. The complex architecture of the LSTM allows it to remember significant patterns and events in time series datasets well. This translates to good performance since the algorithm is able to dissect recurring trends and recall significant prior occurrences to better forecast values for the target variable (turbidity). GRUs also serve as ideal candidates for small datasets since they yield relatively comparable performance to LSTMs while being significantly faster (Chung et al., 2014). Sensors 68 and 22 have extremely close LSTM and GRU error values, but the GRU would have to have lower values across all three metrics columns to stand as a contender for best performance. Due to its rudimentary infrastructure, RNNs performed the worst for every sensor compared to its improved counterparts.

Although each of the three algorithms in this study had extremely narrow differences in their error metrics due to the small size of the dataset (only 1214 rows), the LSTMs performed the best overall, likely due to their complex architecture allowing them to learn more complex functions within this short-term window of data (Shi et al., 2022). The memory benefits of the LSTM may not be fully realized with such a short sequence, but they were still more highly parameterized than the other models. However, this study's findings are limited by the small dataset, and further research may reveal different outcomes with larger datasets (Chung et al., 2014). We hypothesize that a larger dataset would increase the performance gap between the LSTMs and the other two algorithms (GRU and RNN).

While the LSTM performs the best quantitively based on Table 4, we can re-affirm its superior performance by comparing the time series forecasts in Figures 3, 4, and 5, for the outlet sensor of the Stony Clove catchment (01362370). Comparing these plots, we see that Figure 5 (LSTM) has the least difference between its actual turbidity (in blue) and predicted turbidity (orange) indicating the best fit. In other words, we can visually tell that the LSTM model for sensor 70 in Figure 5 would have the lowest error metrics since most of the error metrics are based on differences in actual vs predicted values. Therefore, the lower the gap/distance between the actual and predicted turbidity lines in the time series plots (Figure 3, 4, and 5), the lower the hypothetical error values for the respective plots. The RNN plot (Figure 3) shows the most variation/noise in its predicted turbidity line due to its inability to remember distant past events to scale forecasted values accordingly. We can clearly see the GRU (Figure 4) shows forecasted turbidity peaks (orange line) closest to the actual turbidity peaks (blue line) compared to the other two models. This makes sense because GRUs ideally yield relatively equal performance to LSTMs on small datasets while being more computationally efficient. The LSTM plot (Figure 5) may not hit the highest magnitude for its predicted turbidity peaks but has a better fit between predicted turbidity line and the actual turbidity; thus, explaining its best performance results in Table 4 (sensor 70). Due to the early stopping conditions established in training the models, all our algorithms yielded results with < 20 epochs of training, which boosted our computation times across the board.

Time Series Model Forecast Plots (Sensor 70) – Daily Data

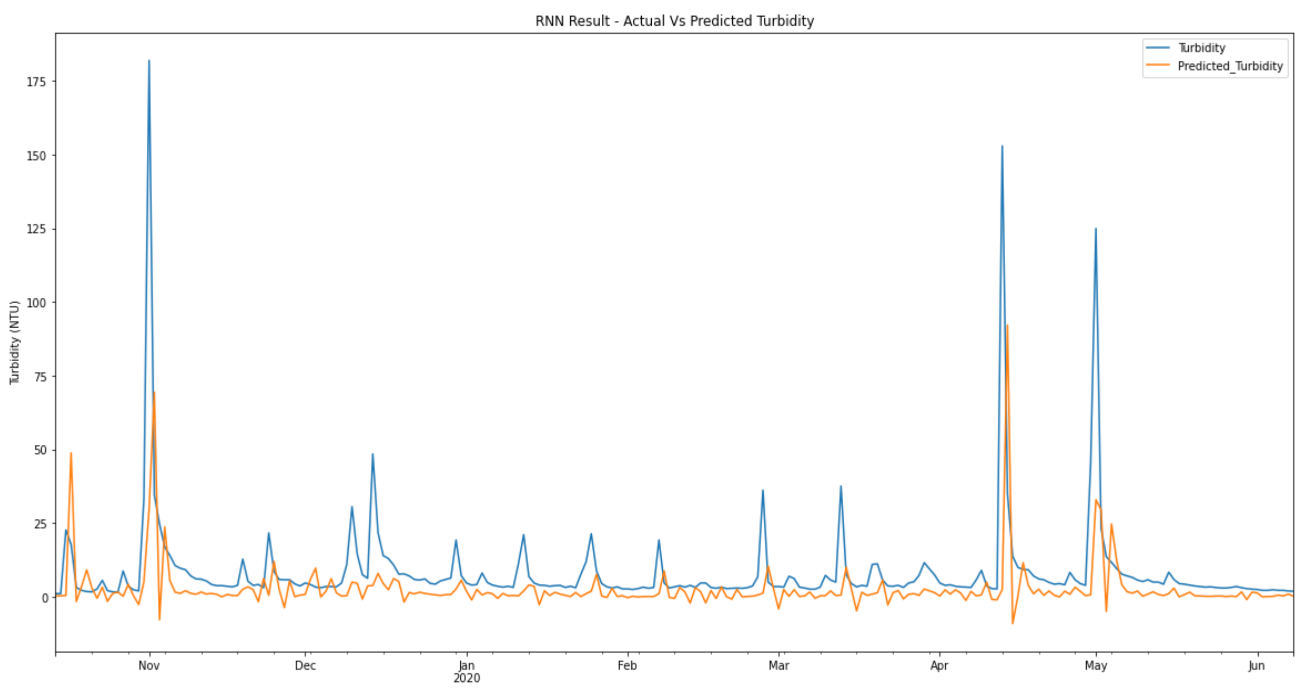

Figure 3. Displays the Actual Turbidity Vs. Forecasted Turbidity plot for the RNN model.

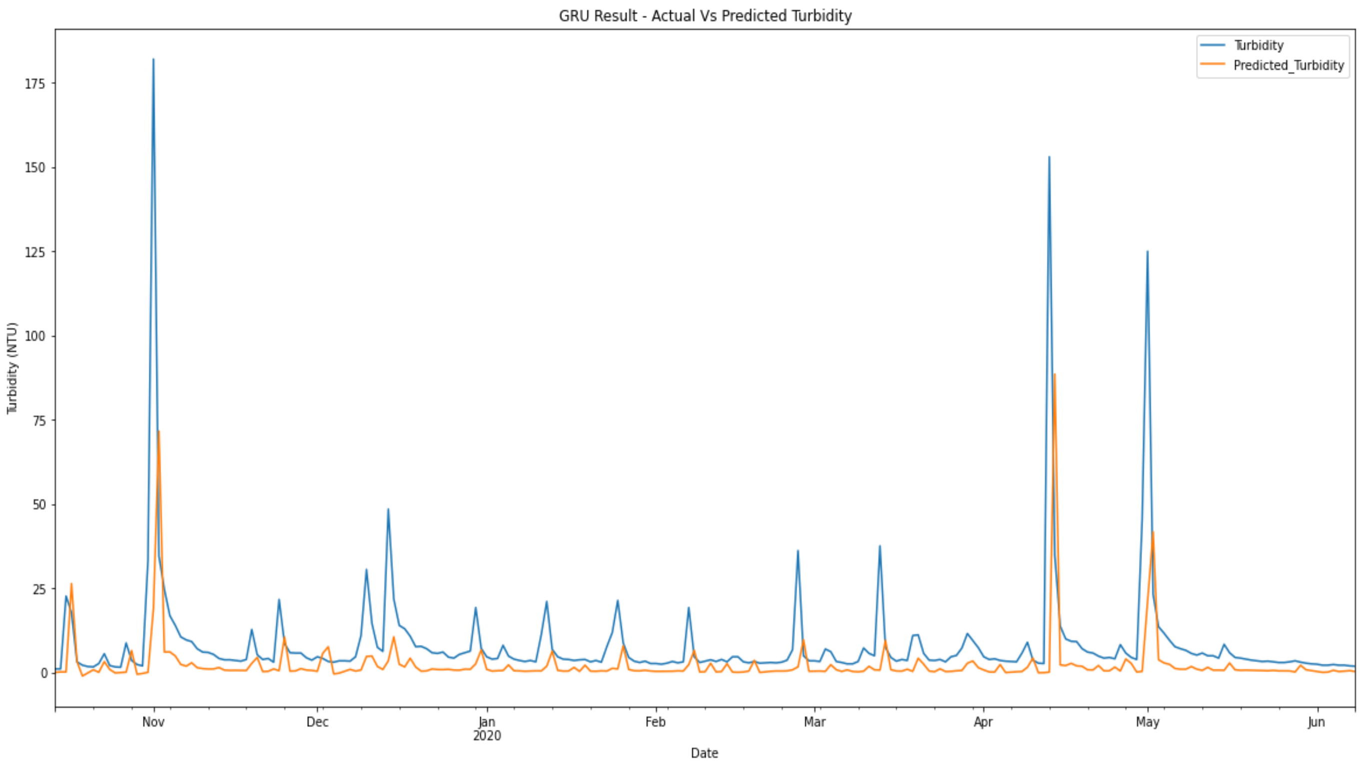

Figure 4. Displays the Actual Turbidity Vs. Forecasted Turbidity plot for the GRU model.

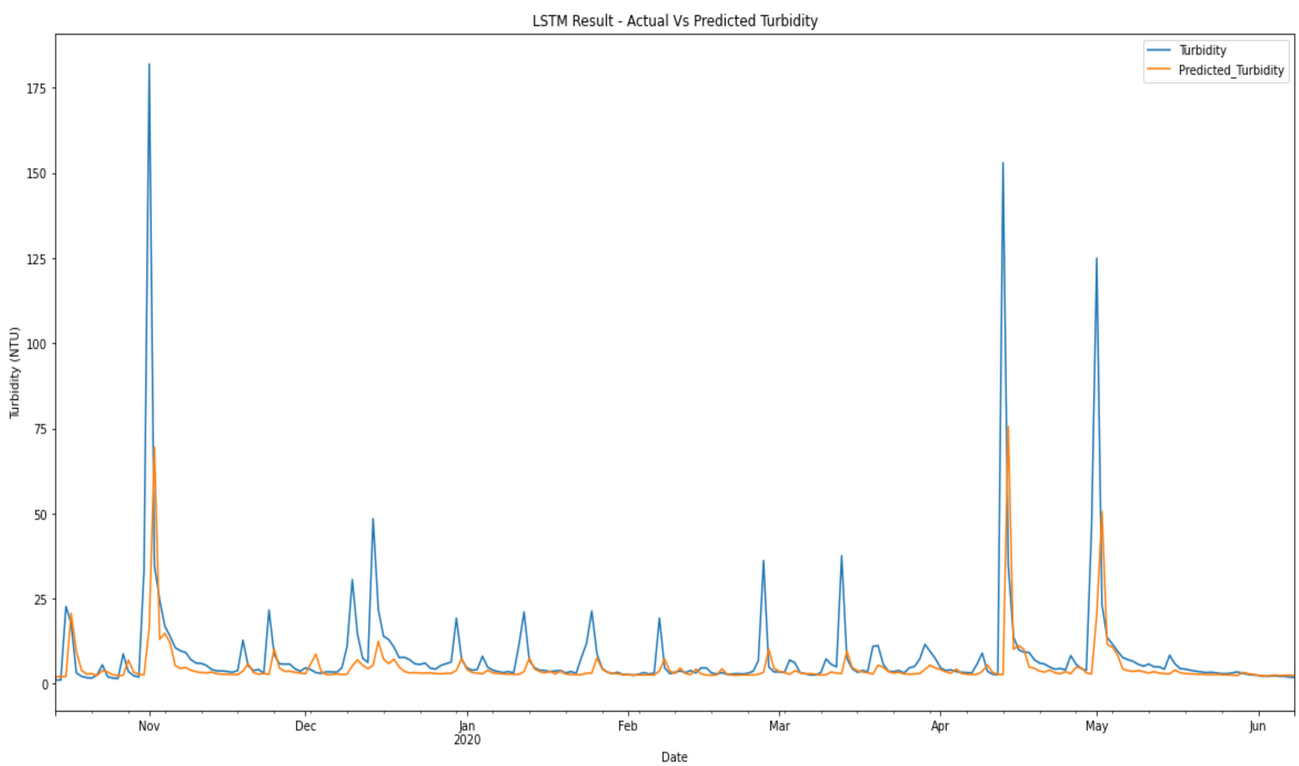

Figure 5. Displays the Actual Turbidity Vs. Forecasted Turbidity plot for the LSTM model.

Table 4. Summarizes algorithm results for each the 5 primary sensor data utilized in this study.
Note – Bold numbers shows best performance for each error metrics (column) per sensor (row)

Sensor	**Algorithm**	**Daily Data Performance Metrics**		
		MAE	MAPE	RMSE
01362370	LSTM	**5.592**	**0.3977**	**17.98**
	GRU	6.991	0.6749	18.44
	RNN	9.595	1.322	19.03
01362368	LSTM	**3.317**	**0.3349**	8.239
	GRU	3.336	0.3372	**8.180**
	RNN	3.659	0.3838	8.346
01362357	LSTM	**5.856**	**0.5483**	**13.71**
	GRU	7.068	0.6279	14.47
	RNN	7.140	0.6430	14.09
01362336	LSTM	**2.230**	**0.2624**	**6.394**
	GRU	3.226	0.4095	6.998
	RNN	3.736	0.5450	7.099
01362322	LSTM	**0.9503**	0.4393	**5.016**
	GRU	0.9930	**0.3913**	5.051
	RNN	1.357	0.8625	5.110

Conclusion & Future Work

Our observed results were in accordance with our initial hypothesis of LSTMs performing better than GRUs and RNNs while yielding similar performance metrics. GRU gives comparable performance to LSTMs on small datasets, which is the case for our current daily-frequency data set, but we expect its performance would decline as we consider higher-frequency data (e.g., 15-minute). The RNN models' performance for every sensor is the poorest compared to its polished counterparts. This can be attributed to the short-term memory issue, vanishing gradients, and inability to ignore noisy/irrelevant data while making its predictions.

Despite the promising predictions thus far, we have numerous improvements and ideas for future work. For example, even though our LSTMs outclassed our RNNs and GRUs for every sensor, we would like to increase the future forecasting horizon to gain more meaningful and realistic outputs from these models. Currently, our goal is to forecast up to 7 days to help NYC DEP optimize reservoir operations if anticipated turbidity levels may spike for any reason such as an upcoming storm event. Future work will also explore algorithms with an alternate and potentially better architecture for addressing complex systems with memory, namely Bi-directional LSTM(BiLSTM) and Transformers. A BiLSTM enhances the capabilities of a Uni-directional LSTM (UniLSTM or regular LSTM) by adding another LSTM layer to the overall structure (Figure 6). Essentially, we have one LSTM layer to feed information or inputs from the forward direction and the other LSTM layer feed inputs from the backward direction. The outputs from both the LSTM layers are combined to make a more pragmatic and accurate forecast of future target variable values. In simple terms, a UniLSTM can only feed and store information from the past to anticipate future values, whereas the BiLSTM feeds inputs from both the past and future values yielding better results (Pirani et al., 2022).

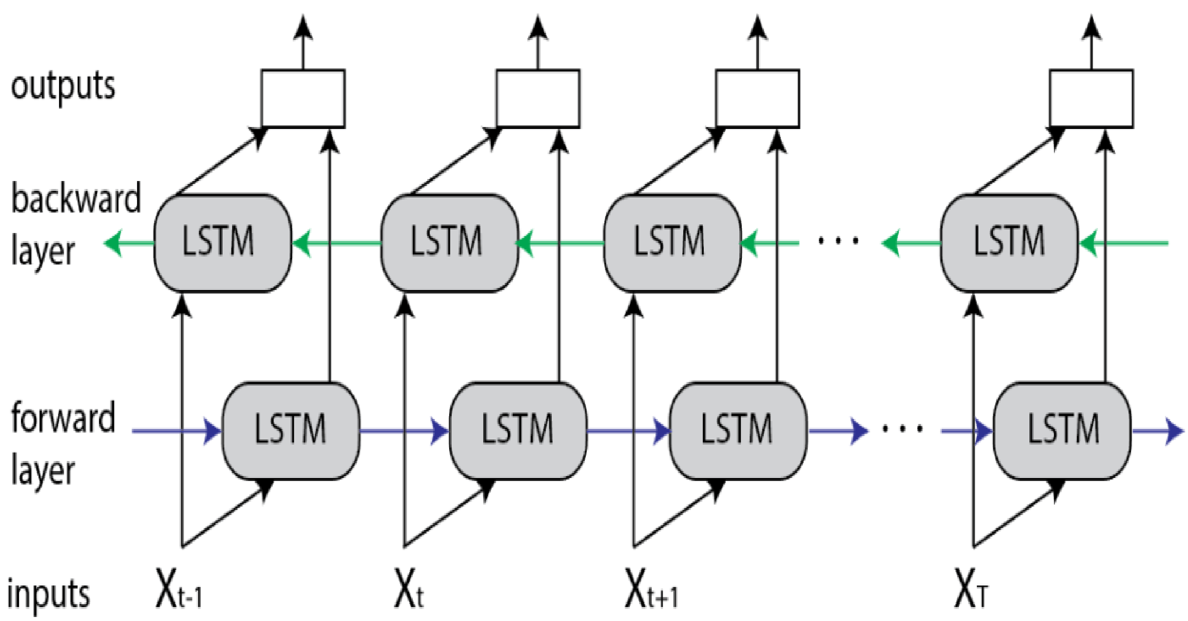

Figure 6. Represents the structure for a Bi-directional LSTM.
Source - https://www.baeldung.com/cs/bidirectional-vs-unidirectional-lstm#:~:text=Bidirectional%20LSTM%20(BiLSTM)%20is%20a,utilizing%20information%20from%20both%20sides.

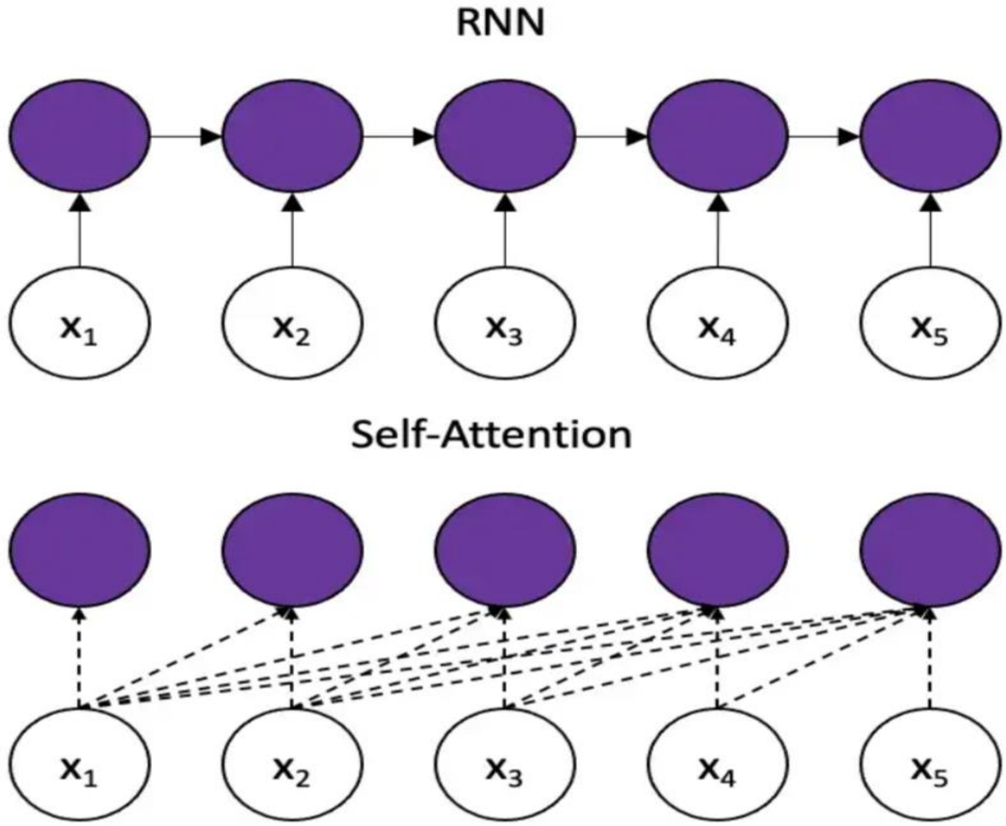

Figure 7. Demonstrates the structural difference between RNNs and Transformers. Source - https://medium.com/mlearning-ai/transformer-implementation-for-time-series-forecasting-a9db2db5c820

A new type of architecture altogether, transformers are making waves in the ML and Deep Learning community by outperforming highly optimized LSTMs. Based on the promising results comparing transformers to RNN approaches (Shi et al., 2022), we wish to build a transformers algorithm to evaluate its performance on this dataset compared to our RNN model approaches. A fundamental difference in the operation of a transformer compared to an RNN is that the recurrent networks tend to process data/token sequentially and lose important information over time due to vanishing gradients (Figure 7). Transformers, on the other hand, establish direct connections to all previous time stamps of data allowing for information to be stored and remembered for almost infinitely long sequences. Another important distinction is that transformers use an algorithm called self-attention to read all input at once (allowing for parallelization) while identifying potentially the most useful sequences to accomplish an assigned task. Another component of our future work involves us exploring the idea of "Nesting" our models together. This means that we would take the predicted results from an upstream sensor and feed it as the input to a downstream station for better overall predictive performance. We hope to modify the architectures of our algorithms to work in this nested format to leverage data from even more sensors and expand our study area to the larger Esopus Creek watershed. Additionally, we hope to expand the scope of our algorithm to other catchments in the Catskill River system to predict turbidity ending up in the Ashokan Reservoir.

ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation under Grant No. NSF-EAR-2012123. Partial funding was also provided to SS through a GUND Barrett Fellowship. I would also like to thank Scott Hamshaw of USGS for his helpful insights and guidance.

Streamflow and Time Series data were downloaded from:

- https://waterdata.usgs.gov/nwis/dv/?site_no=01362370&agency_cd=USGS&%3Breferred_module=sw.

New York State Mesonet data were retrieved February 26, 2022, from:

- https://www2.nysmesonet.org/about/sites#network=nysm&stid=tann

References

Bengio, Y., Simard, P., and Frasconi, P.: Learning long-term dependencies with gradient descent is difficult, IEEE T. Neural Networ., 5, 157–166, 1994.

Brotzge, J. A., Wang, J., Thorncroft, C. D., Joseph, E., Bain, N., Bassill, N., Farruggio, N., Freedman, J. M., Hemker, K., Jr., Johnston, D., Kane, E., McKim, S., Miller, S. D., Minder, J. R., Naple, P., Perez, S., Schwab, J. J., Schwab, M. J., & Sicker, J. (2020). A Technical Overview of the New York State Mesonet Standard Network, *Journal of Atmospheric and Oceanic Technology*, *37*(10), 1827-1845. Retrieved Dec 15, 2022, from https://journals.ametsoc.org/view/journals/atot/37/10/jtech-d-19-0220.1.xml

Chung, Junyoung & Gulcehre, Caglar & Cho, KyungHyun & Bengio, Y.. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.

Daghbandan, A., Khalatbari, S., and Abbasi, M. M. (2019). Applying GMDH-type neural network for modeling and prediction of turbidity and free residual aluminium in drinking water. Desalination & Water Treatment, 140, 118–131. https://doi-org.ezproxy.uvm.edu/10.5004/dwt.2019.23357

Hussain L., Banarjee S., Kumar S., Chaubey A. and Reza M., "Forecasting Time Series Stock Data using Deep Learning Technique in a Distributed Computing Environment," *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, 2018, pp. 489-493, doi: 10.1109/GUCON.2018.8675076.

Iglesias, C., Martínez Torres, J., García Nieto, P., Alonso Fernández, J., Díaz Muñiz, C., Piñeiro, J., & Taboada, J. (2014). Turbidity Prediction in a River Basin by Using Artificial Neural Networks: A Case Study in Northern Spain. Water Resources Management, 28(2), 319–331. https://doi-org.ezproxy.uvm.edu/10.1007/s11269-013-0487-9

Lepot, M., Aubin, J. B., & Clemens, F. H. (2017). Interpolation in time series: An introductive overview of existing methods, their performance criteria and uncertainty assessment. *Water*, *9*(10), 796.

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., and Herrnegger, M.: Rainfall–runoff modelling using Long Short-Term Memory (LSTM) networks, Hydrol. Earth Syst. Sci., 22, 6005–6022, https://doi.org/10.5194/hess-22-6005-2018, 2018.

McHale, M. R., and Siemion, J. 2014. Turbidity and suspended-sediment in the upper Esopus Creek watershed, Ulster County, New York: U.S. Geological Survey Scientific Investigations Report 2014-5200.
Mukundan, R., D. Pierson, E. Schneiderman, D. O'Donnell, S. Pradhanang, M. Zion, et al. 2013. Factors affecting storm event turbidity in a New York City water supply stream. Catena 107:80–88. doi:10.1016/j. catena.2013.02.002
Pirani M., Thakkar P., Jivrani P., Bohara H. M. and Garg D., "A Comparative Analysis of ARIMA, GRU, LSTM and BiLSTM on Financial Time Series Forecasting," 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), 2022, pp. 1-6, doi: 10.1109/ICDCECE53908.2022.9793213.
Sadon N. A., Ismail S., Jafri S. N. and Shaharudin M. S., "Long Short-Term vs Gated Recurrent Unit Recurrent Neural Network For Google Stock Price Prediction," 2021 2nd International Conference on Artificial Intelligence and Data Sciences (AiDAS), 2021, pp. 1-5, doi: 10.1109/AiDAS53897.2021.9574312.
Shi, J.; Jain, M.; Narasimhan, G. Time Series Forecasting (TSF) Using Various Deep Learning Models. *arXiv* 2022, arXiv:2204.11115
Wang, Kezhen & Gelda, Rakesh & Mukundan, Rajith & Steinschneider, Scott. (2021). Inter-model Comparison of Turbidity-Discharge Rating Curves and the Implications for Reservoir Operations Management. JAWRA Journal of the American Water Resources Association. 57. 10.1111/1752-1688.12906.