Wavelet-Based Hybrid Machine Learning Model for Out-of-distribution Internet Traffic Prediction

Sajal Saha, and Anwar Haque
Department of Computer Science
University of Western Ontario, London, ON, Canada
Email:{ssahas9, ahaque32}@uwo.ca

Abstract—Internet traffic prediction is a crucial component for the proactive management of self-organizing networks (SON) to ensure better Quality of Service (QoS) and Quality of Experience (QoE). Modern machine learning techniques have shown outstanding performance in analyzing and predicting complex internet traffic, which has non-linear and non-stationary characteristics. But most existing works assumed that model training and testing data come from independent and identical distribution (IID), which is hardly valid in actual scenarios. Also, they considered synthetic traffic datasets, which do not have enough random properties like real-world traffic. As a result, the model’s prediction accuracy measured using IID data samples is inconsistent with the accuracy of out-of-distribution (OOD) data instances. In this study, we investigated several machine learning models’ performances using four actual traffic datasets whose distribution is different than each other. The best prediction accuracy using IID samples was 96.4% which significantly dropped when we used OOD samples to evaluate the same model. Therefore, we proposed a hybrid machine learning model combining discrete wavelet transformation to decompose original data into several hierarchical components before feeding them into a prediction model. We train our hybrid models using these detail components as features that improve our best performance using IID samples by 1%. Also, it considerably reduces the best accuracy gap of conventional machine learning models in predicting IID and OOD samples by 3.5%, 6.7%, and 2.1%, respectively, for three OOD test sets.

Index Terms—internet traffic, IP traffic prediction, machine learning, out-of-distribution, wavelet decomposition

I. INTRODUCTION

The internet is continuing to change how we connect with others, organize the flow of things, and communicate information across the world. The demand for network traffic has risen significantly around the globe as network technology has advanced and digital activities such as video streaming, remote conferencing, online gaming, and e-commerce have increased. However, predicting network traffic based on historical trends is indispensable for better Quality of Service (QoS), Quality of Experience (QoE), dynamic bandwidth reservation and allocation, congestion control, admission control [1], and privacy-preserving routing [2]. Several research works have been done for efficient and accurate traffic prediction based on conventional statistical models [3] and modern machine or deep learning techniques.

Real internet traffic is a non-linear time series [4], and it is challenging to develop an accurate prediction model due to time-variability, long-term correlation, self-similarity, suddenness, and chaos [5] in internet traffic. Despite those non-linear characteristics, machine learning and deep learning-based methods have impactful performance. Most of the works assumed data is independent and identically distributed (IID), i.e., the train and test data for model development and evaluation came from similar distribution. But in practice, the data distribution will not always be the same due to heterogeneous and anomalous internet traffic. In addition, there is a rising indication that deep learning and machine learning models exploit undesired results due to selection biases, confounding factors, and other biases in the data [6]. These biases in the datasets failed in generalizing the prediction rules for the out-of-distribution data [7]. Also, they assist predictive models in minimizing empirical risk by relying on correlation rather than causality. As a result, it is concerning for real-world Artificial Intelligence (AI) solutions in sensitive domains such as self-driving cars [8], health care [9], etc.

Efficient traffic prediction is crucial for effective business decisions such as infrastructure expansion or abridgment, new service adaption, etc. Machine learning models [10] would be an excellent solution for handling the dynamic nature of internet traffic and providing better predictions. However, the traffic prediction tool is more likely to deal with a data distribution that is slightly different and utterly unknown than the distribution used for model training and testing. Therefore, it is essential to build a robust machine-learning model that will be able to handle a shift in the data distribution. This work proposes a hybrid machine learning model based on discrete wavelet transformation to improve the out-of-distribution performance. The main contributions of our work are as follows:

• Analyzing the performance of several conventional machine learning models using both IID and OOD data samples shows how these classical methods perform for unseen data with different data distributions.

• Proposing a hybrid machine learning model by integrating discrete wavelet decomposition for improving the performance of the state-of-the-art machine learning models using OOD data samples that result in better generalization.

• Evaluating the proposed hybrid architecture using four different real-world traffic datasets that are not identical to the training dataset to show the effectiveness of our proposed methodology for OOD generalization.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
• Investigating different regressor models for single-step traffic prediction. Also, the individual predictor performances have been ensembled using the stacking generalization approach for better accuracy.

This paper is organized as follows. Section II describes the literature review of current traffic prediction using machine learning models. Section III presents the methodology, including data preprocessing, discrete wavelet transformation, machine learning models, and experiment details. Section IV summarizes the different machine learning methods’ performance with both identical and out-of-distribution data and draws a comparative picture between standalone and hybrid models. Finally, section V concludes our paper and sheds light on future research directions.

II. LITERATURE REVIEW

A comparative performance analysis between a conventional statistical model, ARIMA, and a deep learning model, LSTM, has been conducted in [11] for internet traffic prediction. In addition, they used the signal decomposition technique, discrete wavelet transforms (DWT), to separate the linear and non-linear components from the original data before feeding them into the prediction model. The performance of different deep sequence models is investigated in [12]. They treated anomalies in the time series data before feeding into the prediction model and showed the effectiveness of outlier detection in internet traffic prediction. A hybrid model consisting of a statistical and deep learning model is proposed in [13] for better performance than the standalone model. In addition, they applied Discrete Wavelet Transform (DWT) on the time series data for separating the linear and non-linear components, modeled respectively using Auto Regressive Moving Average (ARIMA) and Recurrent Neural Networks (RNN). Since the conventional statistical models, such as ARMA, and ARIMA, are incapable of handling the non-linear components in the time series, the author tried to use the signal decomposition technique here to deal with the complex, non-stationary internet traffic by combining the power of deep learning. According to their experimental results, the combination of ARIMA and RNN performed better than the individual model.

An artificial neural network model combined with the multifractal DWT is proposed in [14]. The network traffic is decomposed into low-frequency and high-frequency components using Haar Wavelet, which has been considered a target for the ANN model with the input of the original traffic data. In the end, model predictions are combined to reconstruct the actual value. Their model outperformed compared with two existing methodologies. In [15], they performed a comparative analysis among different methods of DWT and spline-extrapolation in predicting the characteristics of the IoT multimedia internet traffic. The spline-extrapolation with the B-splines was the best, giving them the minimum forecast error of 5% compared with Haar-wavelet and quadratic splines having prediction errors respectively 7-10% and 10%. In [16], the author developed a traffic prediction framework by combining the power of several technical methods such as Mallat wavelet transformation, Hurst exponent analysis, model parameter optimization, and fusion of multiple prediction models. Firstly, a three-level decomposition has been carried out on the original traffic data to extract a set of approximate and detailed components. Then, the individual component predictability was analyzed using Hurst exponent analysis, where a higher Hurst exponent (H) indicates more predictableness. According to their study, the approximate component has a higher H than the detailed component. As a result, a more efficient machine learning model, the Least squares Support vector machine (LSSVM), is used to predict detail components while the approximate component is analyzed using ARIMA. The proposed method showed better prediction accuracy compared with the other models.

The use of signal decomposition in real-world internet traffic prediction is prevalent in the current literature. The existing works indicate the outperformance of the hybrid model capable of handling the linear and non-linear components separately using different types of models. However, most research assumes that the train data and test data have come from the same distribution, i.e., $P_{tr}(X,Y) = P_{te}(X,Y)$. But, the train and test distribution can be different due to several reasons, such as the temporal/spatial evolution of data or the sample selection bias in the data collection process. Therefore, out-of-distribution (OOD) generalization discusses machine learning methodology where a distribution shift exists between $P_{tr}$ and $P_{te}$. In OOD problem settings, we need to define how the test distribution is different from the strain distribution. There are different distribution shifts in the literature, but the most common one is the covariate shift, where the target generation process is the same with the marginal distribution of X shifts from the training set to the test set. According to recent studies, machine learning methods do not guarantee the generalization of out-of-distribution data. And this issue is not considered extensively in the existing literature on internet traffic prediction. Therefore, it is significantly vital to building a robust prediction model so that it can generalize the unknown distribution in the future. In this research, we propose a hybrid machine learning model that can better generalize the covariate shift in the data.

III. METHODOLOGY

In this section, we describe each component of our proposed traffic prediction model depicted in Fig. 1. Also, we include a detailed description of our experimental datasets used for machine learning model training and evaluation.

A. Training and Testing Dataset

We have used four different real-world internet traffic datasets to conduct our experiment. The datasets are denoted by the letters A, B, C, and D, respectively. The telemetry data was collected by sampling the value of the SNMP (Simple Network Management Protocol) interface MIB (Management Information Base) of a core-facing interface on a provider edge router. Samples are taken at 5-minute intervals, with the bps
Dataset A, B, C, D
Training data
Testing data

Fig. 1. High-level framework of proposed methodology

B. Data Preprocessing

The data preprocessing module of our proposed hybrid traffic prediction framework consists of two tasks: data transformation and normalization and feature extraction. Firstly, we have analyzed the missing values in our datasets. There are 29 missing values in our dataset A, which are replaced using the forward-filling technique where a previous valid data instance has been used to replace the missing value in our traffic data. There are other methods, such as linear interpolation, quadratic interpolation, etc., to handle the missing data in time series analysis. But linear interpolation assumes a linear relationship among data points. It estimates a missing value connecting points in a straight line which is inappropriate in our case as our experimental data is non-linear. Also, the polynomial interpolation method seems unsuitable because we must specify the order before applying this method to replace the missing data. It replaces missing values with the smallest probable degree that passes through available data points. After analyzing different replacement methods, we finally chose the forward-filling technique for handling missing values in our experiment. Then we performed normalization/standardization operations in our training and testing datasets before using them to train and evaluate the prediction models. The normalization step is crucial to avoid biases in model fitting caused by different measurement scales.

Finally, we convert our time series data format, (time, values), into a format of (features, target), which is compatible with supervised learning. A sliding window technique has been used for this conversion process, and we extracted the lagged feature from our time-series dataset. In this method, several previous data samples, i.e., lagged feature, have been used to predict the next value or target in time-series data. However, we must define the window width for this feature extraction process, indicating the number of features. We have used several window-width parameters for our experiment, such as 6, 9, 12, and 15, to extract the optimal input size for the prediction model.

C. Discrete Wavelet Transform

Our proposed hybrid prediction model has a signal transformation component based on wavelet decomposition. We considered different machine learning models for capturing non-linear and non-stationary internet traffic. However, the conventional machine learning model has some limitations, such as model overfitting, unable to find the global optima and poor performance for out-of-distributed data. To overcome those limitations, we integrated a non-classical signal transformation module called Discrete Wavelet Transformation (DWT) to extract hierarchical components of the complex signal representing the detail characteristics. The DWT has the capability to understand the random properties of the non-linear real-world internet traffic, while the classical transformation techniques perform better with linear data.

DWT is a mathematical way of finding the hidden patterns in the original signal by transforming the signal into a time-frequency domain. The process involves a wavelet, i.e., a
wave like oscillation for extracting multiple lower resolution levels by controlling the scale and location of the wavelet [17]. There are different types of wavelets for signal decomposition. However, in our proposed hybrid model, we selected mother wavelets from two major wavelet categories: orthogonal and bi-orthogonal. The orthogonal wavelets filters: low-pass and high-pass filters have the same length as opposed to bi-orthogonal filters. In addition, the orthogonal wavelet filters are symmetric, while the low-pass bi-orthogonal filter is symmetric, and the high-pass filter is either symmetric or anti-symmetric. However, signal transformation properties are strongly correlated with the wavelet characteristics, so we should choose the mother wavelet based on the problem requirements. We utilized and compared three frequently used mother wavelets: dmey (orthogonal), haar (orthogonal), and bior (biorthogonal) in our experiment. These wavelets have been widely used in the existing works for time-series analysis [18], [19].

At each stage of DWT, the signal is decomposed into two components: approximate component, \( Ca \), and detailed component, \( Cd \) representing the general trend and detailed events in the data, respectively. The \( Ca \) from level \( i \) is used to calculate the \( Ca \) in the next level, \( i+1 \). A low-pass filter \( (l_p) \) and high-pass filter \( (h_p) \) convolute the signal to generate the new \( Ca_{i+1} \) and \( Cd_{i+1} \). We used these decomposed components to train our prediction model so that the model can have a better understanding of the individual component of the signal. After predicting the individual component, we can reconstruct the original data by combining all level’s detailed components and approximate components from the last level. We perform experiments with and without this decomposition component to compare the performance of the standalone and hybrid models in predicting testing data that is not identically distributed with the training data. The equation for components and converting them into original data again is as follows [20]:

\[
Ca_{i+1}[n] = Ca_i * l_p[n] = \sum_{m=-\infty}^{\infty} Ca_i[m]l_p[n-2m] \quad (1)
\]

\[
Cd_{i+1}[n] = Ca_i * h_p[n] = \sum_{m=-\infty}^{\infty} Ca_i[m]h_p[n-2m] \quad (2)
\]

Original data, \( y_t = Ca_n + \sum_{i=1}^{n} Cd_i \) \quad (3)

D. Traffic Prediction Model

We have considered boosting algorithms for our traffic prediction task in this work. Boosting is a supervised learning technique for classification and regression problems. It is an ensemble approach that selects the final prediction model based on individual weak learning models. The gradient boosting algorithm’s prediction power is better than individual models and has less problem of data overfitting. They are well-known for their generalization ability, higher prediction accuracy, and better training speed. Therefore, different variations of Gradient boosting algorithms have been considered in this work to evaluate their performance for both in-distribution and out-distribution data samples. In the following subsection, we summarize each type of Gradient boosting algorithm used for our experimentation.

1) Gradient Boosting Regressor (GBR): The fundamental goal of boosting is to successfully combine a large number of weak models in order to produce robust and effective prediction models via greedy searching [21]. Gradient boosting successively fits the decision trees; thus, the fitted trees will learn from the errors of previous trees and thereby reduce the errors. Until the chosen loss function is no longer reduced, more functions are continually added to the already existing ones.

2) Extreme Gradient Boosting (XGB): It carries out the Gradient boosting algorithm that was created in [22]. Compared to other Gradient boosting implementations, this model is relatively quick to compute and may be utilized for classification and regression problems. XGB addresses one of the main inefficiencies of gradient-boosted trees: taking into account the potential loss for every split that could occur during the creation of a new branch. By examining the distribution of features over all data points in a leaf, XGB is able to address these inefficiencies by narrowing the search space for potential feature splits.

3) Light Gradient Boosting Machine (LGB): LGB is another implementation of the Gradient boosting algorithm proposed in [23]. Gradient-based One Side Sampling (GOSS) and Exclusive Feature Bundling (EFB), two unique techniques,
are used in this framework to reduce the restriction of the histogram-based methodology.

4) **Cat Boost Regressor (CBR):** Contrary to other Gradient boosting models, CBR [24] does not follow a similar approach. An index of a leaf can be calculated using bitwise operations because CBR generates oblivious trees, which implies that the trees are formed by enforcing the requirement that all nodes at the same level test the same predictor under the same conditions. The tree structure functions as a regularisation to locate the best solution and prevent overfitting.

5) **Stochastic Gradient Descent (SGD):** It is a straightforward but effective machine learning model used for classification and regression [25]. The amount of samples used to compute the derivatives is the primary distinction between gradient descent and SGD. Compared to gradient descent, SGD dramatically reduces the number of computations by randomly selecting one data sample for each iteration of the gradient calculation.

**E. Evaluation Metrics**

We used Weighted Average Percentage Error (WAPE) to estimate the performance of our traffic forecasting models. The performance metric identifies the deviation of the predicted result from the original data. For example, WAPE error represents the average percentage of fluctuation between the actual value and the predicted value. Therefore, we can define our performance metric mean accuracy (MA) based on WAPE as follow where $p_i$ and $o_i$ are predicted and original value, respectively, and $n$ is the total number of a test instance.

$$WAPE = \frac{\sum_{i=1}^{n} |p_i - o_i|}{\sum_{i=1}^{n} |o_i|} \times 100\%$$  \hspace{1cm} (4)

$$MA = (100 - WAPE)\%$$  \hspace{1cm} (5)

**F. Software and Hardware Preliminaries**

We used Python and machine learning libraries Scikit-Learn [21] and PyWavelets [26] to conduct the experiments. Our computer has the configuration of Intel (R) i3-8130U CPU@2.20GHz, 8GB memory, and a 64-bit Windows operating system.

**IV. RESULTS AND DISCUSSION**

In this section, we analyze our experimental data to find out the result. Firstly, we evaluate the performance of our proposed hybrid model based on dataset A where 70% and 30% data are used for model training and in-distribution testing, respectively. After that, the performance was evaluated using three other datasets, B, C, and D, which are entirely unknown to the model, to measure the effectiveness of our proposed model for out-of-distribution generalization. The performance of all machine learning models trained and validated using dataset A is summarized in Table I.

The group column ‘Standalone (No Wavelet)’ represents the standalone model performance while the other three columns ‘Hybrid (dmey), ‘Hybrid (haar), and ‘Hybrid (bior3.7)’ represent the hybrid model with the corresponding mother wavelet. In addition, we used four different window widths such as 6, 9, 12, and 15, for our feature extraction process. Therefore, all our experimental results summarize the performance of the prediction models based on different feature sets.

**A. Standalone Model Performance Analysis**

In this section, we analyze and compare the performance of our proposed prediction model using data samples from a similar distribution of the training dataset. In the case of the standalone model where the signal decomposition module was off, the best prediction accuracy by the XGB model is 96.1% using the six input features. The highest accuracy for LGB, SGD, GBR, and CBR are respectively 96.2%, 96.2%, 96.2%, and 96.3% using inputs 6, 12, 12, and 6. However, the ensemble model outperformed the individual prediction models and gave us the highest prediction accuracy of 96.4% using 15 input features. Using a stacking ensemble, five different heterogeneous prediction model performances were combined to tune the forecast using logistic regression (LGR). As a result, the ensemble model showed better prediction accuracy than the candidate model.

After evaluating standalone models with a test set having similar distribution as the training set, we used three other datasets from different distributions to validate model performance in case of out-of-distribution. The results are summarized in Table II, Table III, and Table IV, respectively, for dataset B, dataset C, and dataset D. As the distribution of these three datasets differs from the data samples used in training, the standalone model performance decreases significantly in predicting these OOD samples. Fig. 3 depicts a comparative view of best performance by the standalone model and proposed hybrid model. Overall, the standalone model performance was at its peak when it was evaluated using the same distribution data, i.e., training and testing on dataset A. The performance dropped from 96% to 83% when tested using dataset B, where the samples have a distribution shift compared to data samples in A. Also, the performance gap between IID and OOD evaluation using data samples from dataset C and D is significantly large, and it is more than 20% and 10%, respectively. This portion of our experiment indicates a classical out-of-distribution problem of the machine learning model, which assumes both training and testing data are identically distributed.

In our experiment, the classical machine learning model showed a pretty high prediction accuracy of more than 96% for IID samples which indicates their capability of handling non-linear and non-stationary traffic data. But the standalone models were struggling to maintain a similar range of prediction accuracy when tested with OOD data. The accuracy falls from 96% to 75%-86% compared to IID and OOD evaluation. However, there is a high probability of having a distributional shift in the data after deployment of the prediction model in the real world, which we mimicked in our work by considering three different data distributions for evaluating our proposed model. Also, we tried to propose a hybrid machine learning model so that it can reduce the IID and OOD performance...
A is better than the standalone model. The best hybrid model distributed data, i.e., training and testing using both datasets later for converting results into original data. However, the hybrid model is trained using decomposed components instead of original time-series data. This operation allowed the general trend of the traffic compared to the standalone model, it indicates our proposed model better captures the transformation to handling out-of-distributed datasets. For example, the hybrid XGB model with dmey wavelet gave a significant performance enhancement after applying wavelet distribution as the training dataset. Our experiment showed a completely unknown datasets which do not have similar distribution as we trained them using detailed components.

### B. Hybrid Model Performance Analysis

In our hybrid model architecture, the models have an extra module that transforms the original signal into several hierarchical detailed components based on wavelet transformation. The hybrid model is trained using decomposed components instead of original time-series data. This operation allowed the prediction model to learn individual components we aggregate later for converting results into original data. However, the performance of the hybrid model evaluated using identically distributed data, i.e., training and testing using both datasets A is better than the standalone model. The best hybrid model performance we achieved is 97.4% using the ensemble hybrid model (demy), which is a 1% improvement over the standalone model. Next, we evaluated our hybrid model performance using completely unknown datasets which do not have similar distribution as the training dataset. Our experiment showed a significant performance enhancement after applying wavelet transformation to handling out-of-distributed datasets. For example, the hybrid XGB model with dmey wavelet gave us more than 8% accurate results for dataset B compared with the standalone model. Similarly, more than 10% accuracy has been improved in the case of dataset C by the hybrid LGB model with haar wavelet. Overall, the hybrid model’s performance is better than the standalone model for each

### TABLE I

| Standalone (No Wavelet) | Hybrid (demy) | Hybrid (haar) | Hybrid (bior3.7) |
|------------------------|---------------|---------------|-----------------|
| XGB                    | 96.1          | 96.2          | 96.3           |
| LGB                    | 95.9          | 96.1          | 96.2           |
| SGD                    | 96.1          | 96.0          | 96.2           |
| CBR                    | 96.2          | 96.2          | 96.3           |
| ENS                    | 96.2          | 96.0          | 96.1           |

### TABLE II

| Standalone (No Wavelet) | Hybrid (demy) | Hybrid (haar) | Hybrid (bior3.7) |
|------------------------|---------------|---------------|-----------------|
| XGB                    | 78.9          | 83.4          | 79.3           |
| LGB                    | 78.5          | 82.9          | 80.1           |
| SGD                    | 80.1          | 81.6          | 81.7           |
| CBR                    | 80.6          | 83.3          | 82.8           |
| ENS                    | 79.3          | 80.4          | 79.9           |

### TABLE III

| Standalone (No Wavelet) | Hybrid (demy) | Hybrid (haar) | Hybrid (bior3.7) |
|------------------------|---------------|---------------|-----------------|
| XGB                    | 72.6          | 75.6          | 73.9           |
| LGB                    | 77.9          | 75.9          | 73.9           |
| SGD                    | 74.1          | 74.7          | 73.2           |
| CBR                    | 74.0          | 74.4          | 73.7           |
| ENS                    | 73.5          | 74.4          | 73.7           |

### TABLE IV

| Standalone (No Wavelet) | Hybrid (demy) | Hybrid (haar) | Hybrid (bior3.7) |
|------------------------|---------------|---------------|-----------------|
| XGB                    | 85.3          | 84.5          | 80.5           |
| LGB                    | 86.5          | 84.5          | 86.4           |
| SGD                    | 85.0          | 84.5          | 84.5           |
| CBR                    | 86.5          | 86.3          | 86.2           |
| ENS                    | 86.1          | 86.1          | 86.1           |
According to the Fig. 3, the best prediction accuracy for out-of-distributed data has been jumped by more than 4.5%, 7.5%, and 3% respectively for dataset B, C, and D when compared with the standalone model. Also, the hybrid model reduces the best IID and OOD performance gap from 13% to 10%, 20% to 13%, and 10% to 7%, respectively, for dataset B, C, and D compared with the classical machine learning model. The comparison between actual and predicted traffic by best-performing hybrid models for different datasets has been depicted in Fig. 4. Due to the complex and non-linear characteristics of internet traffic, it is challenging to figure out the actual trend and pattern using original time-series data, affecting the standalone model’s performance for unknown distribution. In contrast, the detailed hierarchical component of the time-series data gave more information about the general trend of the traffic.

V. CONCLUSION

We proposed a hybrid machine learning model integrating wavelet decomposition for real-world internet traffic forecast-
ing in this work. Most of the existing methods for traffic prediction are assessed based on the assumption that the training and testing data distribution is the same. But in practice, the prediction model will most likely encounter a slightly different test distribution, especially in the internet domain, as real-world traffic is susceptible to various internal and external factors. However, we considered four datasets having different distributions to build and evaluate our proposed model. The larger dataset is used for model training and in-distribution testing. In contrast, the other three smaller datasets were utilized for out-of-distribution evaluation, which was unknown to the model. Our proposed hybrid model outperformed the individual models for in-distribution and out-of-distribution testing by 1% and 8%-10%, respectively. Due to the distribution shift, the individual model performance decreased significantly for unknown data samples, while the hybrid model analyzed the detailed hierarchical component of the traffic data giving better generalization ability for out-of-distribution. However, the proposed hybrid model suffers from few shortcomings. Firstly, the proper selection of the mother wavelet for data decomposition directly impacts model accuracy. In our work, the wavelets are chosen based on the empirical results of the existing literature. In the future, we would like to extend this work by adapting empirical mode decomposition (EMD) to avoid the problem of choosing a suitable wavelet for data decomposition. We will also apply optimization methods for selecting an appropriate mother wavelet based on entropy and energy. Secondly, the DWT has the problem of circular shift, where a tiny change in signal origin highly affects the output, and it requires a dyadic length dataset for decomposition. To overcome these shortcomings, we would like to explore a modified DWT approach called Maximal Overlap Discrete Wavelet Transform (MODWT), which is circular shift-invariant and unrestricted by the dyadic length constraint. Finally, we plan to extend our experiment with multi-step forecasting as it is more challenging and beneficial than single-step prediction.

REFERENCES

[1] H. Kalibkhani, M. G. Shayesteh, and N. Haghighat, “Adaptive Istar model for long-range variable bit rate video traffic prediction,” IEEE Transactions on Multimedia, vol. 19, no. 5, pp. 999–1014, 2016.

[2] G. Han, H. Wang, X. Miao, L. Liu, J. Jiang, and Y. Peng, “A dynamic multipath scheme for protecting source-destination privacy using multiple sinks in wsns intended for iot,” IEEE Transactions on Industrial Informatics, vol. 16, no. 8, pp. 5527–5538, 2019.

[3] S. Saha, A. Haque, and G. Sidebottom, “An empirical study on internet traffic prediction using statistical rolling model,” in 2022 International Wireless Communications and Mobile Computing (IWCMC), 2022, pp. 1058–1063.

[4] D. Wei, “Network traffic prediction based on rbf neural network optimized by improved gravitation search algorithm,” Journal of Network and Computer Applications, vol. 28, no. 8, pp. 2303–2312, 2017.

[5] Q. Li and R.-C. Liu, “A new approach for chaotic time series prediction using recurrent neural network,” Mathematical Problems in Engineering, vol. 2016, 2016.

[6] Z. Shen, J. Liu, Y. He, X. Zhang, R. Xu, H. Yu, and P. Cui, “Towards out-of-distribution generalization: A survey,” arXiv preprint arXiv:2108.13624, 2021.

[7] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio, “Toward causal representation learning,” Proceedings of the IEEE, vol. 109, no. 5, pp. 612–634, 2021.

[8] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixao, F. Mutz et al., “Self-driving cars: A survey,” Expert Systems with Applications, vol. 165, p. 113816, 2021.

[9] A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun et al., “Scalable and accurate deep learning with electronic health records,” NPJ Digital Medicine, vol. 1, no. 1, pp. 1–10, 2018.

[10] S. Saha, A. Haque, and G. Sidebottom, “Towards an ensemble regressor model for isp traffic prediction with anomaly detection and mitigation,” in 2022 International Symposium on Networks, Computers and Communications (ISNCC), 2022, pp. 1–6.

[11] G. N. N. Barbosa, M. A. Lopez, D. S. V. Medeiros, and D. M. F. Mattos, “An entropy-based hybrid mechanism for large-scale wireless network traffic prediction,” in 2021 International Symposium on Networks, Computers and Communications (ISNCC), 2021, pp. 1–6.

[12] S. Saha, A. Haque, and G. Sidebottom, “Deep sequence modeling for anomalous isp traffic prediction,” in ICC 2022 - IEEE International Conference on Communications, 2022, pp. 5439–5444.

[13] R. Madan and P. S. Mangipudi, “Predicting computer network traffic: A time series forecasting approach using dwt, arima and rm,” in 2018 Eleventh International Conference on Contemporary Computing (IC3). IEEE, 2018, pp. 1–5.

[14] Y. Lu, H. Li, B. Lu, Y. Zhao, D. Wang, X. Gong, and X. Wei, “Network traffic model with multi-fractal discrete wavelet transform in power telecommunication access networks,” in International Conference on Simulation Tools and Techniques. Springer, 2019, pp. 53–62.

[15] I. Strelkovskaya, I. Solovskaya, A. Makoganiuk, and T. Rodionova, “Multimedia traffic prediction based on wavelet-and-spline extrapolation,” in 2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom). IEEE, 2020, pp. 1–5.

[16] Z. Tian, “Network traffic prediction method based on wavelet transform and multiple models fusion,” International Journal of Communication Systems, vol. 33, no. 11, p. e4415, 2020.

[17] Z. Liu and L. Menzel, “Identifying long-term variations in vegetation and climatic variables and their scale-dependent relationships: A case study in southwest germany,” Global and Planetary Change, vol. 147, pp. 54–66, 2016.

[18] Y. Yeo, S. Kim, O. Kisi, and V. P. Singh, “Daily water level forecasting using wavelet decomposition and artificial intelligence techniques,” Journal of Hydrology, vol. 520, pp. 224–243, 2015.

[19] A. Kumar, R. Komaragiri, and M. Kumar, “Time–frequency localization using three-tap biorthogonal wavelet filter bank for electrocardiogram compressions,” Biomedical engineering letters, vol. 9, no. 3, pp. 407–414, 2019.

[20] S. O. Mousavizadeh Kashi and M. Akbarzadeh, “A framework for short-term traffic flow forecasting using the combination of wavelet transformation and artificial neural networks,” Journal of Intelligent Transportation Systems, vol. 23, no. 1, pp. 60–71, 2019.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[22] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.

[23] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, “Lightgbm: A highly efficient gradient boosting decision tree,” Advances in neural information processing systems, vol. 30, 2017.

[24] L. Prokhorenkova, G. Gusev, A. Vorobiev, A. V. Dorogush, and A. Gulin, “Catboost: unbiased boosting with categorical features,” Advances in neural information processing systems, vol. 31, 2018.

[25] D. Saad, “Online algorithms and stochastic approximations,” Online Learning, vol. 5, no. 3, p. 6, 1998.

[26] G. Lee, R. Gommers, F. Waselewski, K. Wohlfahrt, and A. O’Leary, “Pywavelets: A python package for wavelet analysis,” Journal of Open Source Software, vol. 4, no. 36, p. 1237, 2019.