MC-ANN: A Mixture Clustering-Based Attention Neural Network for Time Series Forecasting

Yanhong Li, *Graduate Student Member, IEEE*, and David C. Anastasiu, *Member, IEEE*

***Abstract*—Time Series Forecasting (TSF) has been researched extensively, yet predicting time series with big variances and extreme events remains a challenging problem. Extreme events in reservoirs occur rarely but tend to cause huge problems, e.g., flooding entire towns or neighborhoods, which makes accurate reservoir water level prediction exceedingly important. In this work, we develop a novel extreme-adaptive forecasting approach to accommodate the big variance in hydrologic datasets. We model the time series data distribution as a mixture of both point-wise and segment-wise Gaussian distributions. In particular, we develop a novel End-To-End Mixture Clustering Attention Neural Network (MC-ANN) model for univariate time series forecasting, which we show is able to predict future reservoir water levels effectively. MC-ANN consists of two modules: 1) a grouped Auto-Encoder-based Forecaster (AEF) and 2) a mixture clustering-based learnable Weights Attention Network (WAN) with an attention mechanism. The WAN component is crucial, skillfully adjusting weights to distinguish data with varying distributions, enabling each AEF to concentrate on clusters of data with similar characteristics. Through extensive experiments on real-world datasets, we show MC-ANN's effectiveness (10–45% root mean square error reductions over state-of-the-art methods), underlining its notable potential for practical applications in univariate, skewed, long-term time series prediction tasks.**

***Index Terms*—Pattern recognition, deep recurrent neural networks, encoder, decoder, time series forecasting, attention clustering network, Gaussian mixture, hydrology prediction, reservoir water level.**

I. Introduction

TIME Series Forecasting (TSF) has been researched extensively, since it has many real-world applications, including weather prediction [1], stock market value prediction [2], and agricultural management [3], among others. In hydrologic prediction, complex unpredictable factors like weather, geography, and human activity also affect the water level of dams and reservoirs. Reservoirs are multipurpose, vast bodies of water that play a vital role in flood control, navigation, irrigation, and energy production. They also have a direct impact on human safety and welfare. Extensive study on reservoir water level prediction has been prompted by their versatile use cases. By serving as organic drainage networks, reservoirs control precipitation and snowmelt runoff, maintaining ecological balance and preventing flooding. However, substantial seasonal shifts cause significant fluctuations in water levels and the non-stationary and big variance character of these changes makes traditional forecasting models less accurate.

Traditional machine learning models usually consider seasonality, long-term patterns, and non-stationary properties when making predictions based on these data, yet predicting time series with big variances and extreme events remains a challenging task. Despite deep learning previously being used to predict reservoir water levels, its effectiveness is often limited to individual basins and fails to capture non-stationary patterns with high variance. Moreover, current deep learning models [4], [5], [6] struggle with time series data that includes abrupt changes or rare yet critical extreme events, particularly in nonstationary reservoir datasets characterized by extreme values such as the ones we focus on in this work.

Various studies seek to decode the nonlinear nature of time series through statistical methods. Mixture density networks [7] are used for multi-modal data where each modality can be captured using mixing components [8]. Klotz et al. [9] examine the uncertainty in streamflow predictions using mixture density networks and Monte Carlo Dropout. However, their potential to enhance water level predictions has not been explored, especially in terms of simultaneously accounting for long-term trends in a single forecast and updating short-term predictions in a rolling fashion. In a previous work [10], we introduced NEC+, a probability-enhanced composite model featuring three distinct components—a predictor for normal values, another for extreme events, and a classifier to merge the two—tailored to enhance prediction accuracy for hydrology time series that include extreme events. This model, however, is not an end-to-end model and its performance highly depends on the accuracy of its classifier component.

These challenges have lead us to explore how we can design an end-to-end neural network capable of utilizing distinct statistical features to enhance the predictive performance of non-stationary time series with big variance and extreme events, which we present in this work. Our contributions include:

- We develop a novel end-to-end Mixture Clustering Attention Neural Network (MC-ANN) for univariate time series forecasting, which effectively addresses reservoir water level time series prediction.
- The MC-ANN model incorporates a novel clustering-based importance enhanced sampling strategy that adeptly pinpoints critical features and trends within datasets, thereby

Received 3 March 2024; revised 12 March 2025; accepted 23 April 2025. Date of publication 28 April 2025; date of current version 3 July 2025. Recommended for acceptance by J. Wang. *(Corresponding author: David C. Anastasiu.)*

The authors are with the Computer Science, Engineering, Santa Clara University, Santa Clara, CA 95053-4345 USA (e-mail: yli20@scu.edu; danastasiu@scu.edu).

Digital Object Identifier 10.1109/TPAMI.2025.3565224

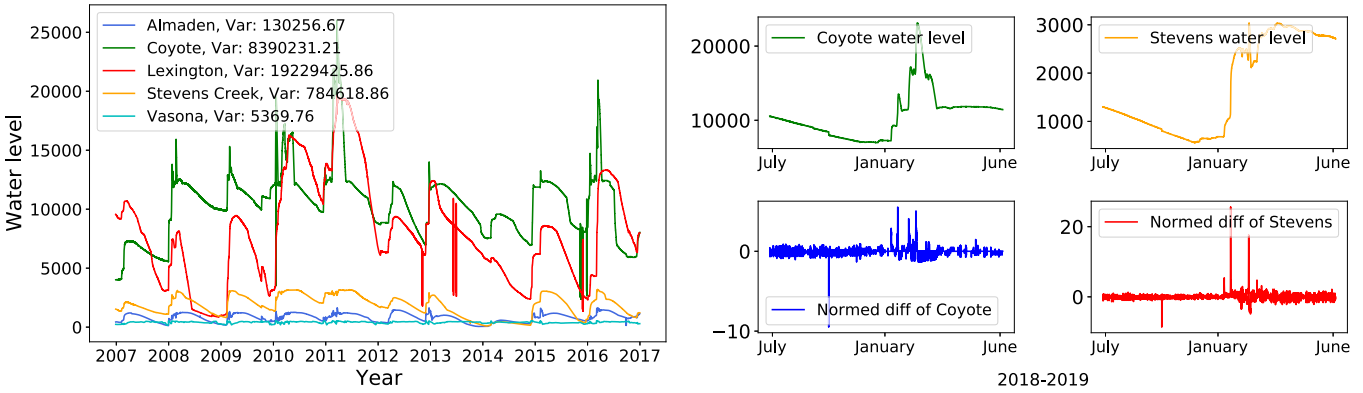

Fig. 1. Water level for the five reservoirs in our study from 2007 to 2017 (left). For the Coyote and Stevens Creek reservoirs, water level data from July 2018 to June 2019 show significant variance (right). Upon applying first-order differencing and normalization to the data, extreme values become evident in the normalized change of the water level series.

TABLE I
RESERVOIR SENSOR ORIGINAL DATA AND FIRST-ORDER DATA STATISTICS

Sensor	Min	Max	Variance	Skewness	Kurtosis
Original data					
Almaden	-0.10	1714.08	1714.08	-0.27	-0.76
Coyote	2319.42	27421.25	13304899.11	1.30	3.33
Lexington	867.88	20109.10	21374524.46	0.50	-0.36
Stevens Creek	93.92	3229.98	713429.64	-0.20	-0.93
Vasona	140.15	619.48	5044.46	-1.05	0.67
First-order difference data					
Almaden	-408.51	417.11	16.13	0.10	3541.97
Coyote	-10262.42	11782.86	5865.46	6.10	8985.19
Lexington	-10852.95	7841.56	4122.16	-18.42	11104.88
Stevens Creek	-1395.84	1092.97	49.75	-22.78	21075.51
Vasona	-90.29	74.39	2.45	3.41	438.38

enhancing forecasting precision by relying on the learned mixture distribution of the data.

- Our model performs exceptionally well in real-world circumstances, providing the best results when performing rolling predictions throughout the year, while also outperforming baselines in the single-shot 3 d-ahead prediction task.
- Through an extensive set of experiments on five real-world datasets (Table I, Fig. 1), we have shown that our model performs well (10%—45% of root mean square error reductions over state-of-the-art methods) for reservoir time series with big variance and extreme events.
- The model integrates a distinctive attention-based Loss, blending point-wise and segment-wise clustering approaches, which has been validated in ablation studies to enhance accuracy by upwards of 20%.

Beyond academic performance, accurate forecasts have direct real-world impacts. In agriculture, precise water level predictions support optimized irrigation, enhancing crop yield sustainability. In hydropower, improved forecasting enables better reservoir management, preventing operational inefficiencies. In flood management, reducing forecast errors can strengthen early warning systems, helping to mitigate risks. Our model is deployed in real-world applications. It has been integrated into FlowView, a web-based platform that provides Santa Clara Valley Water District with real-time 3-day ahead stream flow and water reservoir level predictions. This system aids in decision-making for reservoir operations, water distribution, and flood risk assessment.

II. RELATED WORK

A. Machine Learning Methods

The prediction of time series has been studied for many years. Traditional machine learning methods that were once widely used include the univariate Autoregressive (AR), Moving Average (MA), Simple Exponential Smoothing (SES), and Extreme Learning Machine (ELM) algorithms; however, the Autoregressive Integrated Moving Average (ARIMA) [11] method and its various variations were the most well-known. Wang et al. [12] presented a hybrid model for long-term streamflow forecasting that combines Ensemble Empirical Mode Decomposition (EEMD), Empirical Mode Decomposition (EMD), and ARIMA. Several studies employed Gaussian Process Regression (GPR) [13] and Quantile Regression (QR) [14] to measure forecast uncertainty in addition to making predictions. Tree-based models have also been used because they are computationally efficient and can handle predictors without assuming any particular distribution. Examples of these models are classification and regression trees (CART) and random forests (RF). A popular time series forecasting model, Prophet [15], is based on an additive model that incorporates seasonal and holiday influences at many time scales, including annual, weekly, and daily patterns and captures nonlinear trends in the data. However, machine learning techniques are hindered by unequal data distribution, as demonstrated by Singh et al. [16], who also pointed out that balancing the dataset is a crucial step in the training process.

B. Deep Learning Methods

Deep learning models have recently become the method of choice for predicting rich time series data [17], surpassing traditional statistical methods like GARCH [18] and ARIMA. The NBeats method proposed by Oreshkin et al. [4] outperformed all

competitors on the standard M3, M4, and TOURISM time series datasets, demonstrating good performance on general time series prediction [5]. To anticipate future values and their confidence, DeepAR [5] assumes a conditional distribution over the future values and trains a shared RNN to predict them. TimesNet [19] transforms 1D time series data into a 2D representation and utilizes receptive fields for prediction. Designed for general time series analysis, it discards high-frequency signals as noise during the transformation process, which differs from our objective of capturing extreme values.

Transformer-based techniques, such as Autoformer [6] and Reformer [20], have been proposed recently to address the problem of long-term forecasting by granting the transformer with more sophisticated dependency discovery and modeling capabilities. A ProbSparse self-attention mechanism and a generative style decoder were shown to work well in Informer [21], which significantly increases the inference speed of long-sequence predictions. However, new research has cast doubt on these models' effectiveness, suggesting that more straightforward linear models may perform better [22], [23]. In response, some studies explore novel ways to leverage Transformers for time series forecasting. PatchTST [24] segments time series into patches and applies a channel-independent Transformer, enhancing efficiency and long-term forecasting accuracy. iTransformer [25] embeds each time series as variate tokens, using attention to capture correlations among channels and a feed-forward network for series representations. It applies Transformer components on inverted dimensions to enhance forecasting performance.

C. Extreme Adaptive Methods

Despite extensive research in time series prediction, deep learning models face difficulties when dealing with time series data that contain rare or extreme occurrences because of the obvious imbalance in the dataset. This calls for the creation of specialist models intended for precise forecasting of extreme events.

An and Cho [26] proposed a novel method that focused on anomaly detection using reconstruction probability as a lens. This approach cleverly takes into account the data distribution's intrinsic variability. Similar to this, the Uber TSF model [27] automatically extracts extra features from the auto-encoder LSTM network, priming it to capture intricate time-series dynamics during large-scale events. New inputs are then fed to the LSTM forecaster for prediction. By employing the Generalized Extreme Value Loss (GEVL), Zhang et al. [28] took into account different heavy-tailed distribution kernels (Gumbel, Frechet) for loss estimation, which transforms the loss estimator to a heavy-tailed distribution. The variational disentangled extremal (VIE) classifier [29] model uses representation learning and a combination of Gaussian and Generalized Pareto distribution priors to classify data with extreme events. Yifan et al. [30] proposed a framework to integrate machine learning models with anomaly detection algorithms, where the extreme events are highlighted so the machine learning models can process them appropriately. Additionally, we previously proposed NEC+ [10], a model specifically designed to provide good prediction performance on hydrologic time series with extreme events. By employing the Gaussian mixture model (GMM) [31] and training three predictors in parallel, our model was able to achieve the best forecasting performance for reservoir water level time series with rare but important extreme events, without sacrificing the quality of normal values prediction.

D. Limitations of Current Approaches

Based on previous research, a variety of neural network designs, including recurrent neural networks [32], [33], hybrid networks [4], and graph neural networks [34], [35], [36] have been examined for hydrologic forecasting. An ensemble LSTM and Prophet model developed by Du and Liang [37] was demonstrated to perform better than any of the individual models employed in the ensemble. To tackle the hydrologic prediction problem, Le et al. [38] combined an encoder-decoder architecture with an attention mechanism [39], [40]. For the reservoir water level forecasting problem, Ibañez et al. [41] looked at two variants of the LSTM-based DNN model: a multivariate version (DNN-M) and a univariate encoder-decoder model (DNN-U). Trigonometric time series encoding was utilized in both models. To build an extreme adaptive predictor, some work used Extreme Value Theory (EVT) [30] and probability enhanced methods [10]. However, very few of these earlier studies have focused on end-to-end handling of both long-term sequences and extreme events in reservoir water level forecasting, especially in the more practical scenario of rolling prediction.

To bridge this gap, we develop a novel End-To-End Mixture Clustering Attention Neural Network (MC-ANN) for univariate time series forecasting, which addresses effective prediction of long-term time series with big variances and extreme events. Remarkably, MC-ANN's performance surpasses that of state-of-the-art methods across five real-world reservoir datasets, in both the task of single-shot 3-day ahead prediction and when considering rolling predictions every 8 hours.

III. Preliminaries

A. Problem Statement

We address a demanding univariate time series forecasting challenge, dealing with a non-stationary series characterized by significant variance and containing extreme events. The data includes both extreme declines and rapid increases in value. While the majority of typical fluctuations are key to the overall prediction accuracy, the critical task is to precisely forecast the infrequent yet severe changes to prevent potential disasters. The problem can be described as,

$$[x_1, x_2, \ldots, x_T] \in \mathbb{R}^T \rightarrow [x_{T+1}, \ldots, x_{T+H}], \in \mathbb{R}^H$$

i.e., we are predicting the vector of length-H horizon future values given a length-T observed series history, where x_1 to x_T are the inputs and x_{T+1} to x_{T+H} are the outputs. Root mean square error (RMSE) and mean absolute percentage error (MAPE), as standard scale-free metrics, are used to evaluate

forecasting performance. Our predictive models utilize the past 15 days of hourly data (with $T = 360$ time steps) to predict water levels 72 hours ahead ($H = 72$), covering a forecast period of three days.

B. Data Statistics

For our study, we utilized an extensive dataset comprising 31 years of hourly water level readings from reservoir sensors in Santa Clara County. These reservoirs, constructed in the 1930 s and 1950 s, are designed for water conservation, capturing stormwater runoff that would otherwise flow into the San Francisco Bay. Besides aiding in flood control and offering recreational activities, the reservoirs support environmental health by ensuring river flows are maintained.

Detailed information about the sensor locations and their corresponding statistics can be found in Table I. Throughout the paper, we will reference the sensors and their time series data by the specific locations listed in the table. To provide a more detailed perspective, Table I presents several computed statistics for our input time series, including minimum, maximum, variance, skewness, and kurtosis. The original water level data, detailed in the upper section of the table, exhibit exceptionally large variances due to seasonal changes.

Time series exhibiting trends or seasonality are inherently non-stationary and typically result in suboptimal predictions. To counter this, we employ first-order differencing, a common preprocessing technique in time series analysis used to achieve stationarity, which subtracts the previous value from the current value for each point in the time series, i.e., $x'_t = x_t - x_{t-1}$. The lower section of Table I displays the first-order difference of the original data, reflecting water level variations.

High skewness and kurtosis values of the first-order difference dataset indicate considerable asymmetry and deviation from the symmetrical bell curve typical of a normal distribution, as discussed in Chissom's "Interpretation of the kurtosis statistic" [42]. The negative skewness in a distribution indicates a leftward bias, with a longer tail on the lower end, which suggests a higher frequency of extreme decreases in water levels. In contrast, positive skewness is indicative of rapid increases presented as positive extreme values.

C. GMM Decomposition

A GMM [31] is a non-supervised clustering method that assumes data is derived from a mixture of several Gaussian distributions, each representing a cluster within the complex overall distribution. GMM is particularly adept at disentangling these distributions without the need for labeled training data. The GMM approach estimates the parameters of each Gaussian using the Expectation-Maximization algorithm, iteratively improving the model until it converges on a solution that best explains the hidden, latent structure of the data. This makes GMMs powerful for uncovering subgroups in multi-modal data and understanding the intricate distributions within a dataset. GMM can be described by the equation,

$$p(x|\lambda) = \sum_{i=1}^{M} w_i \; g(x|\mu_i, \mathbf{\Sigma_i}), \tag{1}$$

where x is a D-dimensional continuous-valued vector, $w_i \; \forall i = 1, \ldots, M$ are the mixture weights, and $g(x|\mu_i, \Sigma_i)$ are the component Gaussian densities. Each component density is a D-variate Gaussian function, and the overall GMM model is a weighted sum of M component Gaussian densities,

$$g(x|\mu_\mathbf{i}, \mathbf{\Sigma_i}) = \frac{1}{2\pi^{\frac{D}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right\}, \tag{2}$$

where μ_i is the mean vector and Σ_i is the covariance matrix of the ith component. The mixture weights are constrained such that $\sum_{i=1}^{M} w_i = 1$. Due to its capacity to represent a vast class of sample distributions, GMMs are frequently employed in biometric systems, most notably in speaker recognition systems. The GMM's capacity to produce smooth approximations to arbitrarily shaped densities is one of its most impressive features [31].

In our model, GMMs generate both point-wise and segment-wise probability series, capturing distinct temporal patterns in the data. These probability series serve as inputs to the Weight Adjustment Network (WAN), an attention-based mechanism that dynamically adjusts the contribution of different AutoEncoder-based Forecasters (AEFs). This enables specialized forecasters to focus on normal conditions or extreme events, improving overall prediction accuracy. Through this adaptive weighting, our model enhances robustness in non-stationary reservoir datasets with high variance and extreme values.

D. Rolling Prediction

During the inference phase, we employed a strategy of rolling predictions, forecasting water levels every 8 hours. Each prediction cycle generated 72 future data points, effectively covering water level forecasts for the ensuing 3 days at hourly intervals. These forecasts are based on the preceding 360 time steps, i.e., 15 days in the data. While three-day forecasts can provide a short-term trend of water levels at a given moment, rolling predictions are more commonly used in practice. Continuously updating forecasts are more meaningful for decision-making, which is why our project focuses on the performance of rolling predictions over an annual cycle. Hence, our goal is to achieve superior rolling prediction results without compromising the accuracy of the three-day forecasts.

IV. Methodology

A. MC-ANN Framework

The MC-ANN model initiates by learning the distribution of time series data through a one-dimensional mixture of Gaussian distributions, producing the WGMM features. These WGMM features, along with the preprocessed input, is then used as input for three Auto-Encoder based Forecasters (AEFs), as depicted on the left side of Fig. 2. On the figure's right side, we illustrate the learning of three point-wise clustering feature groups from the preprocessed input series x, and three segment-wise groups from the WGMM feature series g. These are then amalgamated according to a specified policy to create mixed-clustering groups, which serve as input to the Weighted Attention Network (WAN) to specifically disentangle and emphasize extreme values. The

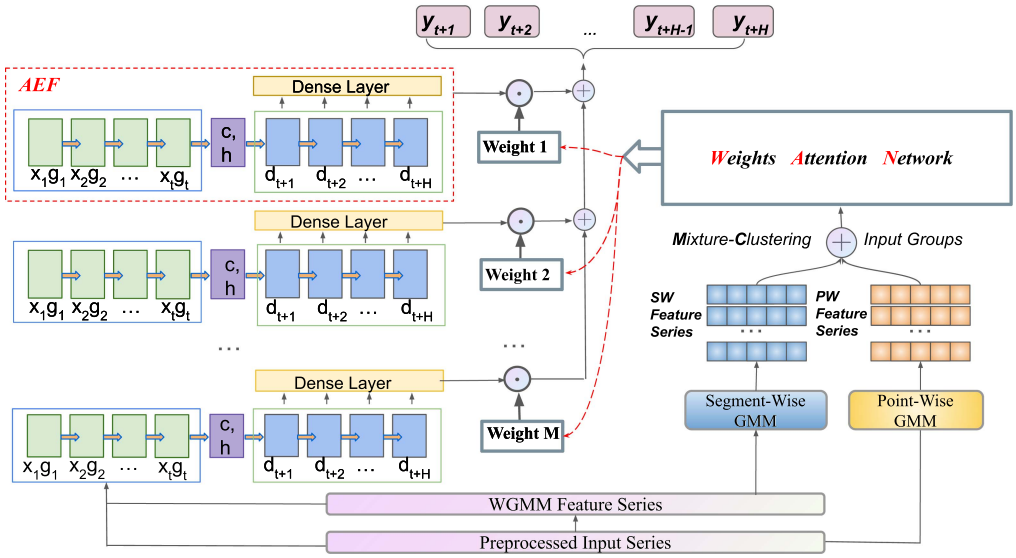

Fig. 2. MC-ANN learns the time series data distribution, as a mixture of Gaussian distributions on both point-wise and segment-wise levels, consisting of two parts: 1) Grouped Auto-Encoder based Forecaster (AEF) and 2) GMM mixture clustering-based learnable Weights Attention Network (WAN) for disentangling extreme values from normal ones. The AEF Forecaster models predict part of the regression values and WAN generates the weights that allow the AEFs to focus on different time series trends.

AEF Forecaster models output partially precise values, while the WAN computes weights to guide the AEFs to concentrate on various scenarios. In the following, we will further explain each of the framework components.

B. WGMM Features

Given the training set input data, we first find a univariate GMM model with M components that best fits the data. The model aims to reconstruct each time series value as a weighted combination of M Gaussian predictors. Then, given the preprocessed input series, we calculate a WGMM feature value for each data point x_j as the weighted sum of the probabilities from all components, i.e., $g_j = \sum_{i=1}^{M} w_i \; g(x_j|\mu_i, \mathbf{\Sigma_i})$. In the middle section of Fig. 3, the higher values in the WGMM feature series, represented by the blue line, indicate a greater likelihood of a data point being an extreme value.

C. Point-Wise Clustering (PW GMM) Feature Series

In addition to generating the WGMM features, we use the trained GMM models to generate M more series that contain only the probabilities of each GMM component for the input, which we call the point-wise clustering (PW) feature series. Each data point in the input series is associated with M potential cluster memberships, each corresponding to one of the GMM components. For instance, the point-wise Gaussian Mixture clustering features for the Coyote reservoir sensor are illustrated in the bottom portion of Fig. 3, where we have highlighted a specific period to better demonstrate the distribution characteristics. It shows that point-wise (PW GMM) clustering features indicate the likelihood of data points belonging to specific clusters. We use the last 72 points from these M series as input to train our Weights Attention Network.

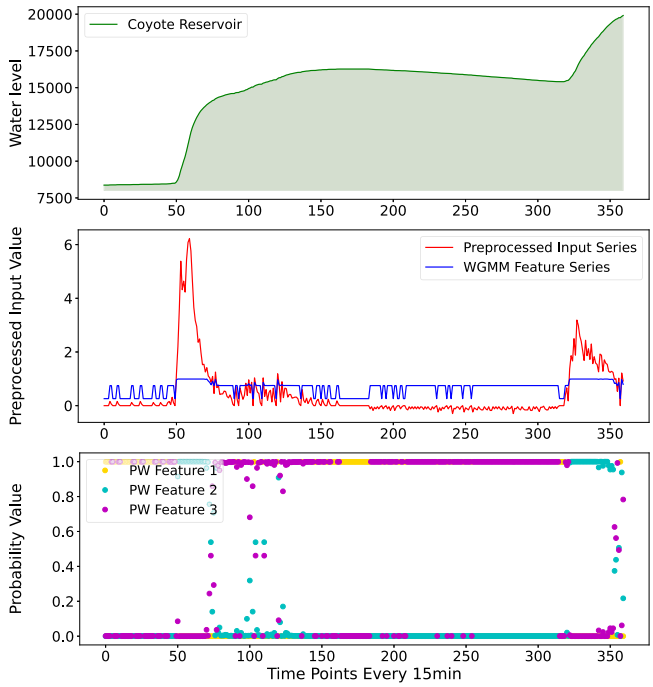

Fig. 3. Point-wise clustering feature series: initial water levels (top), values after preprocessing and the WGMM feature series (middle), and PW features series for a GMM model with $M = 3$ components (bottom).

D. Segment-Wise Clustering (SW GMM) Feature Series

The M PW GMM series described in the previous section contain the probabilities of each of the last 72 input values belonging to each of the M GMM clusters. In general, one of the clusters will contain the most extreme values, and the mixture weight of its associated GMM cluster (w_i from (1)) will likely

be low, since the majority of the points in the time series are not extreme points. Assuming few extreme points in our input, the PW series associated with that GMM cluster will have few high probabilities (close to one) and many probabilities close to zero. Similarly, another cluster may contain values close to the mode of the time series and its GMM mixture weight will likely be much higher. The PW series for this GMM cluster will likely have much higher values overall.

While the PW series capture the point-wise trend in the data, we also wish to capture the trends or shapes of the series. To do so, we train an additional multivariate GMM model with M components that uses 72-point sub-segments from the training time series as inputs. The clusters learned by this model will conform to series with similar shapes rather than similar point values. Moreover, one of the clusters will have many samples with the most prevalent shapes and associated relatively high weight value, while others will have more infrequent shapes, like those we might see when extreme events develop in the series. For a given input sequence in our data, we use the last 72 points in the series to compute the probabilities with witch the series belongs to each of the M multivariate GMM clusters. Then, we form M 72-point segment-wise (SW GMM) clustering feature series by repeating the probabilities 72 times. The SW GMM sequences will be combined with the PW GMM sequences according to the mixture clustering policy we describe in the following section.

E. Mixture Clustering Policy

We have generated M sets of point-wise clustering feature series, as well as M sets of segment-wise clustering feature series, each of length 72. As shown in Fig. 3, point-wise features are valuable for predicting future values due to their representation of data point distributions across clusters. However, there is a notable imbalance issue—few points are associated with the cluster marked in yellow, while the majority are associated with the cluster marked in magenta.

To address this imbalance, we form M composite clusters that are then processed by our Weighted Attention Network (WAN) as separate *Mixture Clustering (MC) Input Groups*. This inclusion introduces a new dimension of learning based on the characteristics of both point-wise and segment-wise clusters. By creating a policy to blend segment-wise and point-wise features, we form three composite clusters that helps WAN discern the optimal weights that will be used to combine predictions for M Auto-Encoder-based Forecasters, enabling our set of AEFs to focus on the most relevant cluster interactions for predictive analysis.

To integrate the three sets of point-wise and segment-wise clustering feature series, we rely on the weights learned when training the respective GMM models. In particular, we sort the M point-wise cluster series in ascending order based on their GMM mixture weights, and we sort the M segment-wise cluster series in descending order based on their respective GMM mixture weights. Each sorted point-wise cluster series is then combined with its corresponding segment-wise cluster series

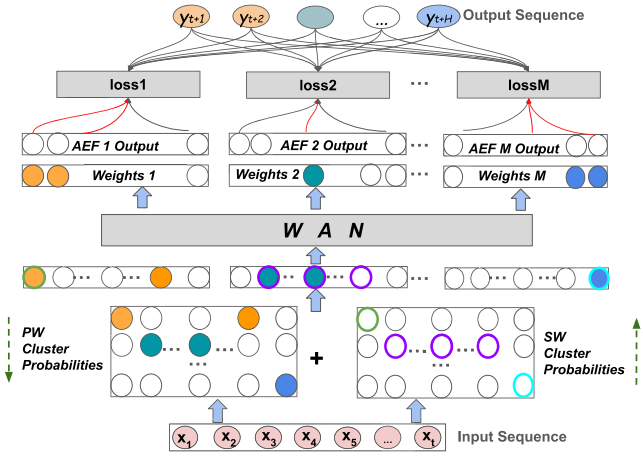

Fig. 4. Attention weights-guided backpropagation.

multiplied by a factor SW_{factor}, such that

$$MC_{Input} = PW_{Series} + SW_{Series} \times SW_{factor}. \quad (3)$$

This results in M mixed-clustering feature series that act as inputs to the WAN.

F. Attention-Based Component Weighting

Fig. 4 shows how the WAN employs an attention mechanism on the combined probability matrix derived from both point-wise and segment-wise clusters to generate loss weights. This enables the WAN to learn a distinct weight vector for each mixed cluster, encapsulating the predictive attention directed towards that particular cluster. These weight vectors are then applied to the outputs of the AEF, contributing to the loss computation and influencing the final prediction. Namely,

$$\mathcal{L}_i = \hat{AEF}_i \odot Weights_i, \ \forall\, i \in \{1, 2, \ldots, M\}, \quad (4)$$

$$\mathcal{L} = RMSE\left(\sum_{i=1}^{M} \mathcal{L}_i, y\right). \quad (5)$$

This process encourages AEFs to concentrate on the unique features of the predicted locations associated with specific cluster characteristics, thereby simplifying the complexity of the prediction task by breaking it down into more manageable components.

G. Weights Attention Network

Leveraging the clustering features from the M identified clusters, we construct a Weights Attention Network that utilizes mixed *point-wise* and *segment-wise* clustering feature series as inputs. This network guides the model to concentrate on learning the distinct facets of the data concurrently, effectively allowing it to distinguish between various data behaviors and patterns.

Our WAN network is composed using M sub models to process the M input mixture cluster groups. Each sub model processes input data through a series of layers, each comprising

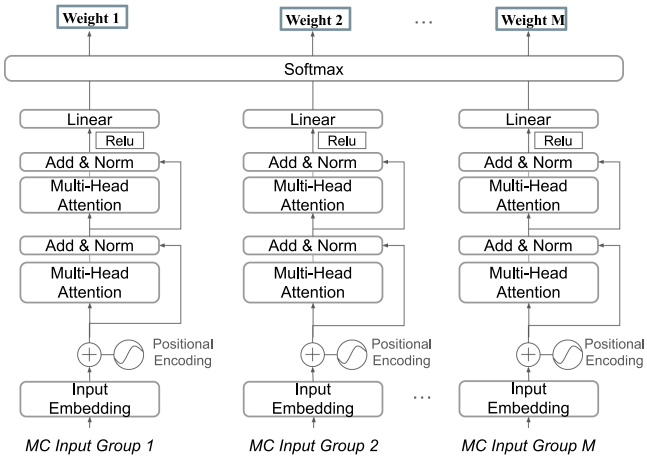

Fig. 5. The **W**eights **A**ttention **N**etwork.

two core components: a stack of multi-head self-attention mechanisms that allow the model to consider the influence of different parts of the input sequence when encoding a specific element, and a feed-forward network that applies a set of linear transformations. Input tokens are first converted into vectors using embedding layers, and, to account for the lack of inherent sequence processing, positional encoding is added to these embeddings. Residual connections follow each sub-layer, facilitating gradient flow during training, while layer normalization is applied to stabilize the network's output. Rectified Linear Unit (ReLU) activation is applied after the linear layer inputs to enhance the model's capacity for capturing non-linear characteristics within the data.

As shown in Fig. 5, we apply the softmax function to the M output vectors from our model, transforming them into probability values. These probability vectors then serve as weight factors, which are used to perform element-wise multiplication with the outputs of the Auto-Encoder based Forecasters (AEF), effectively weighting the contribution of each forecaster's output.

H. Auto-Encoder-Based Forecasters (AEF)

The M forecasters in our model adhere to the original sequence-to-sequence (seq2seq) architecture, as proposed by Sutskever et al. [43]. As depicted in Fig. 2, the encoder, which is an LSTM network, receives past water level observations (x) and the corresponding WGMM feature variables (g). It processes this information to produce a hidden state (h) and a cell state (c). These states from the encoder are subsequently utilized as the initial states for the LSTM decoder. This decoder is initialized with known future dates (d), which correspond to the timestamps for which we aim to predict the water levels. The outputs from the decoder LSTM are then directed through a time-distributed dense layer, tasked with producing the final water level forecasts.

Before providing them as inputs to the decoder, we encode each date, represented initially as the day in the year, into a pair of features $d_t = [d_{sin}, d_{cos}]$ using sine and cosine transformations, also known as trigonometric or cyclical encoding. This method captures the 365-day periodicity inherent in calendar dates, representing them as a pair of values within the range of -1 to 1, thus maintaining the cyclical nature of time within the model.

I. Clustering-Based Oversampling Policy

We design a novel oversampling approach that aims to capture important data points in the sequence based on predefined thresholds and sample data points in their vicinity with specific step size (s) and frequency (ν).

As shown in Section IV-C, GMM is employed to cluster the data into M clusters with mean values $\mu_1, \mu_2, \ldots, \mu_M$. Two threshold values, z_1, and z_2, are determined by taking the means of the top-2 highest and top-2 lowest cluster means, respectively. Samples in the sequence are identified as important if the maximum value in any part of the sequence exceeds z_1 or if the minimum value falls below z_2. For each case, the parameters s and ν are treated as hyperparameters, providing flexibility for fine-tuning the sampling strategy to match the specific characteristics and distribution of the data. By adjusting s and ν, practitioners can effectively control the density and granularity of oversampled points around important regions, ensuring a more targeted and data-driven approach to oversampling. To be specific, the starting point is shifted left of the identified maximum or minimum by $s \times \nu/2$, and then data points are sampled every s points, repeated ν times, effectively yielding $s \times \nu$ samples around each identified maximum or minimum point.

We will select a ratio os to manage the overall volume of the training set. The oversampling process will end when the amount of oversampled data exceeds os percent of the training set. We employ grid search to determine the optimal os value.

V. Evaluation

A. Experimental Settings

1) Dataset: Our dataset[1] includes over 31 years of hourly water level sensor readings for 5 reservoirs in Santa Clara County, CA, which are Almaden, Coyote, Lexington, Stevens Creek, and Vasona, named after their locations and described in Table I. The table shows several statistics for the 5 reservoir sensors used in our study. As an illustration, Fig. 6 shows Stevens Reservoir, an artificial lake located in the foothills of the Santa Cruz mountains, near Cupertino, California. Information about these reservoirs can be found at https://www.valleywater.org/your-water/local-dams-and-reservoirs. Although the majority of the data was accessible starting in 1973, we noticed a large number of missing and anomalous data points in the series' early years as a result of sensor or data storage malfunctions. Therefore, we restricted our study to the years 1991–2019. An adaptive polynomial interpolation method was used to fill in short gaps in the time series during these periods. This method involved projecting the missing values onto a polynomial function that was learned to best match k values before and after a gap of size $2k$.

[1]Data and code for MC-ANN can be found at https://github.com/davidanastasiu/mcann.

Fig. 6. Stevens Creek reservoir, an artificial lake located in the foothills of the Santa Cruz mountains, near Cupertino, California.

Our objective was to forecast the reservoir water level for a year, from July 2018 to June 2019. The training and validation datasets were randomly selected from time series data covering the period from January 1991 to June 2018. Our pre-processing regimen for model training included a first-order differencing, $x_i = x_{i+1} - x_i$ for each time step i, and normalization of the data by standardization (subtract the mean and divide by standard deviation). Post-processing steps to revert the data to its original scale involved reversing the standardization and first-order differencing transformations applied during pre-processing.

2) Model Parameters: In our study, we employ the Expectation-Maximization algorithm to fit the GMM models to the training set input data. Furthermore, we set the number of components M to 3 for each time series. This decision is based on our observations that increasing the number of clusters beyond this tends to produce additional components with diminishing weights, which do not contribute significantly to the model's performance.

In our AEF models, we employed a four-layer Long Short-Term Memory (LSTM) network architecture followed by a hyperbolic tangent (tanh) activation function to normalize the LSTM outputs. Upon evaluating various layer widths—256, 384, 512, 698, and 1024 nodes—we determined that layers with 512 nodes delivered optimal results. For regularization, we implemented a dropout rate of 0.1 to prevent overfitting and promote model generalizability. While $f = 72$ (3 days) was set by our problem definition, we tested $h \in 72, 168, 360, 720$, i.e., 3, 7, 15, and 30 days, and found $h = 360$ to work the best for all reservoirs. Post-LSTM, our model uses a fully connected layer to output the final predictions of AEF.

In the WAN network described in Fig. 5, 384 hidden nodes are used in the attention layer and dense layer for the Almaden, Stevens Creek, and Vasona sensors, and 300 for the Coyote and Lexington sensors.

For simplicity, we set the ν and s oversampling parameters to 18 and 4, respectively, in all our experiments. Essentially, this means that we will oversample the whole prediction portion of a given significant sample every four steps, resulting in an excess of 18 data points around a peak or low point. We used 20% oversampling for the Stevens Creek dataset and 40% for all others. Additionally, we used $PW_{factor} = 0.4$ for the Coyote, Stevens Creek, and Vasona sensors, while the Almaden and Lexington sensors rely solely on SW clustering ($PW_{factor} = 0$).

We utilized the Adam optimizer with an initial learning rate of 0.001, which decays by a factor of 0.9 after each epoch. The training process is configured to run for a maximum of 100 epochs, with early stopping set to trigger after 4 epochs without improvement. All the models involved in the experiments were trained on a dataset comprising 40,000 samples, and we validate their performance using a randomly selected set of 60 samples which were excluded from the training set.

B. Baseline Methods

Our method, MC-ANN, was benchmarked against a comprehensive selection of cutting-edge models used in both general time series and specialized *hydrologic* prediction, as detailed in the related work section. These models include:

- FEDFormer [44], which optimizes the Transformer model by incorporating seasonal-trend decomposition, enhancing efficiency and performance in long-term forecasting,
- iTransformer [25], which applies the attention and feed-forward network on the inverted dimensions,
- Informer [21], a Transformer variant designed for extended time series predictions, featuring a prob-sparse self-attention mechanism,
- NLinear [23], a robust linear approach employing first-order differencing, tailored for long-term time series analysis,
- DLinear [23], a model that applies trend decomposition to improve long-term time series forecasting,
- N-BEATS [4], renowned for its superior performance on several benchmark datasets,
- DNN-U [41], a state-of-the-art univariate LSTM-based encoder-decoder *hydrologic* model used to predict reservoir lagged water levels,
- A-LSTM [38], a state-of-the-art *hydrologic* model used to predict stream-flow by applying attention mechanism to generate the hidden states of a decoder, and
- NEC+ [10], our previous *hydrologic* time series prediction model especially designed for data involving extreme events that employs a suite of LSTM-based models.

C. Main Results

The experiment results are shown in Tables II and III. The best value of each metric is shown in bold. The second-best value is underlined. While we also tested FEDFormer [44], its performance was generally worse than the other transformer-based methods and we leave its results out from the table due to lack of space. Overall, without sacrificing the 3-day prediction accuracy, MC-ANN consistently surpasses other models, with improvements ranging from around 10% to nearly 45% in rolling prediction. Transformer-based methods such as iTransformer, FEDFormer, and Informer, struggle significantly with datasets that exhibit large variances. In the cases of the Coyote, Lexington, and Stevens Creek reservoirs, these methods underperformed markedly, indicating a vulnerability to the challenges posed by substantial fluctuations in the data. Fully connected

TABLE II
EFFECTIVENESS COMPARISONS AGAINST STATE-OF-THE-ART METHODS ON ROLLING 8-HOUR PREDICTION

Datasets	Metric	MC-ANN	NEC+	iTransformer	Informer	NLinear	DLinear	NBEATS	DNN-U	A-LSTM
Almaden	RMSE	**7.412**	10.580	65.683	211.241	16.199	23.009	23.229	9.676	18.040
	MAPE	**0.002**	**0.002**	0.016	0.204	0.005	0.006	0.009	0.004	0.016
Coyote	RMSE	**45.373**	64.590	755.083	7437.162	385.546	346.678	159.024	117.000	1282.913
	MAPE	**0.002**	**0.002**	0.020	0.653	0.013	0.012	0.006	0.004	0.126
Lexington	RMSE	**255.739**	303.510	1600.916	9565.245	645.141	798.529	468.829	318.024	660.354
	MAPE	**0.003**	**0.003**	0.048	0.773	0.023	0.024	0.011	0.005	0.068
Stevens Creek	RMSE	**7.382**	14.977	48.256	714.468	27.380	48.692	34.998	13.363	117.497
	MAPE	**0.002**	**0.002**	0.0136	0.589	0.005	0.012	0.008	0.006	0.104
Vasona	RMSE	**5.137**	5.775	15.269	19.580	7.045	12.544	10.572	11.370	23.587
	MAPE	**0.004**	**0.004**	0.020	0.028	0.006	0.013	0.011	0.016	0.049

TABLE III
EFFECTIVENESS COMPARISONS AGAINST STATE-OF-THE-ART METHODS ON 3-DAYS PREDICTION

Datasets	Metric	MC-ANN	NEC+	iTransformer	Informer	NLinear	DLinear	NBEATS	DNN-U	A-LSTM
Almaden	RMSE	**53.539**	58.117	59.272	217.641	60.516	64.596	64.764	58.648	57.649
	MAPE	**0.014**	**0.014**	0.015	0.162	0.017	0.021	0.018	0.017	0.021
Coyote	RMSE	433.571	466.276	608.228	8507.417	631.056	730.057	535.886	**417.24**	1338.622
	MAPE	**0.011**	**0.011**	0.015	0.619	0.019	0.022	0.013	0.012	0.128
Lexington	RMSE	**774.209**	794.842	926.294	11878.486	1019.081	1082.898	931.356	832.329	1050.5
	MAPE	0.015	**0.014**	0.020	0.930	0.030	0.033	0.023	0.018	0.078
Stevens Creek	RMSE	**71.303**	91.810	93.042	1052.549	90.084	99.598	89.918	76.090	156.591
	MAPE	0.013	**0.011**	0.015	0.888	0.014	0.024	0.015	0.016	0.128
Vasona	RMSE	**18.474**	20.893	18.264	22.051	20.157	20.021	21.405	20.683	32.245
	MAPE	**0.018**	0.020	0.020	0.031	0.020	0.024	0.023	0.027	0.062

networks such as NLinear, DLinear, and NBEATS, which benefit from decomposing data into main trends and residuals, exhibit better performance compared to Transformer-based methods. However, they still fall short of achieving high accuracy. On the other hand, RNN-based models like DNN-U, A-LSTM, and NEC+, which are specifically tailored for hydrologic data forecasting, perform better than the aforementioned methods. Despite this, they do not surpass the effectiveness of MC-ANN, which outperforms all these models in predicting water levels with big variances.

1) 8-Hour Rolling Prediction: Across various reservoirs, the MC-ANN model demonstrates superior performance with substantial improvements in RMSE compared to other forecasting methods. For Almaden, MC-ANN shows an improvement of approximately 23.4% over the next best model, while in Coyote, it achieves a 29.8% better RMSE. Lexington sees a 15.7% improvement, and notably, Stevens Creek and Vasona reservoirs exhibit improvements of 44.8% and 11.1%, respectively. These results underscore MC-ANN's significant advancements in predictive accuracy for rolling water level predictions, highlighting its potential for practical applications in hydrologic forecasting.

2) 3-Days Prediction: In the realm of single-shot 3-day predictions, MC-ANN consistently manifests high performance, marked by notable improvements over various state-of-the-art methods. Overall, MC-ANN performs best on 4 of 5 reservoirs and is in the second position, only slightly behind the leader, in the fifth. Comparing its results here with those in the 8-hour rolling prediction task, we note that MC-ANN excels at predicting the start of the series, consistently outperforming its competition without much trade-off in effectively predicting the remaining time points.

3) Statistical Significance Testing: To validate the robustness of our improvements, we conducted a Wilcoxon Signed-Rank Test comparing MC-ANN with each baseline using RMSE from both rolling and 3-day-ahead predictions across five datasets (10 RMSE values per model). The results confirm that MC-ANN significantly outperforms all baselines ($p < 0.05$ in all cases), showing the statistical significance of our improvements.

4) Inference Examples: Fig. 7 shows several example predictions from the five reservoirs. For visual clarity and to emphasize comparative effectiveness, we only show a few of the closest baseline models in addition to MC-ANN and exclude those with significantly poorer performance. This approach highlights the relative strengths and improvements of MC-ANN over models with the closest accuracy, thereby providing a more focused and impactful visual representation of its predictive capabilities.

Results illustrate that MC-ANN excels in capturing the short-term trends of the data over a 3-day horizon, effectively predicting the nuances of the water level time series. MC-ANN is proficient in tracking rising curves, responding to downward trends, and navigating through fluctuating patterns. Even in instances of large variations, MC-ANN demonstrates its capability to predict these significant changes with great accuracy.

D. Ablation Studies

1) Effects of Cluster-Based Oversampling: Given the data's pronounced skewness and kurtosis, which signal a deviation from a normal distribution, it is impractical to set the

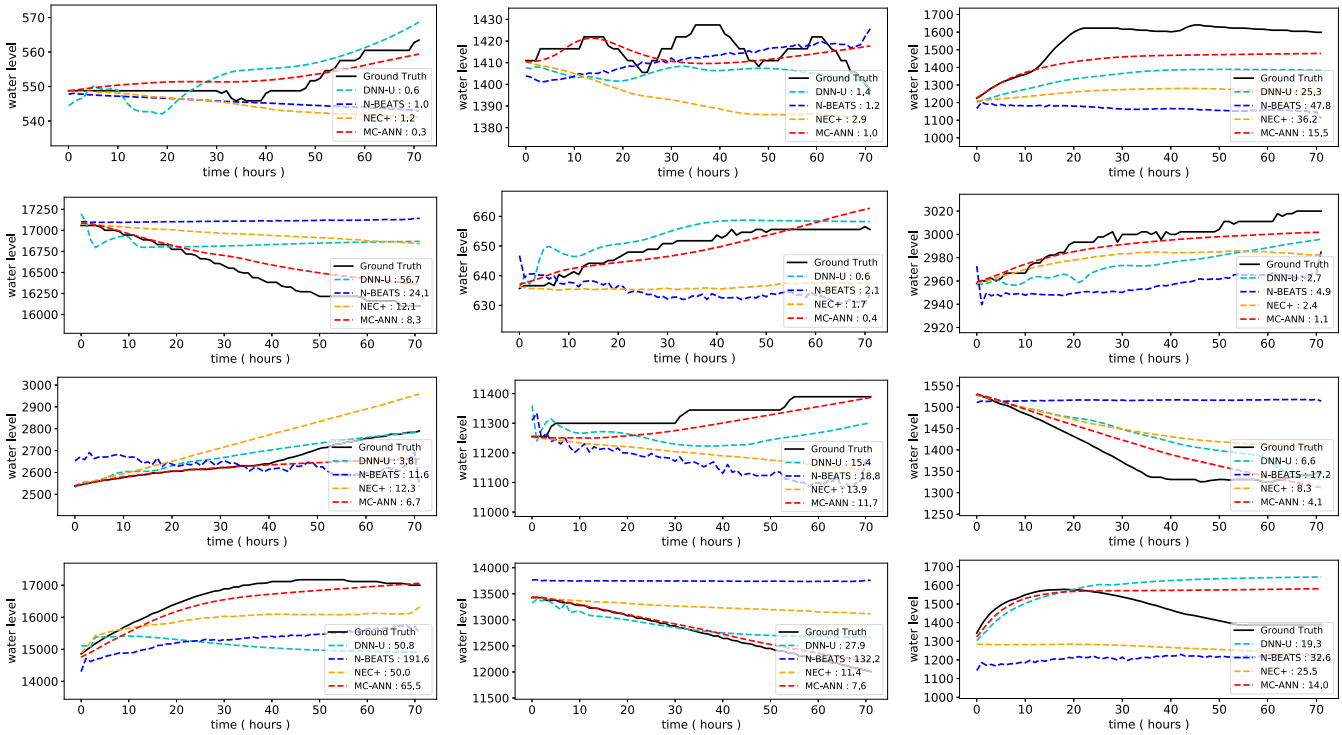

Fig. 7. Effectiveness comparison of MC-ANN and baselines. RMSE values are denoted next to each label.

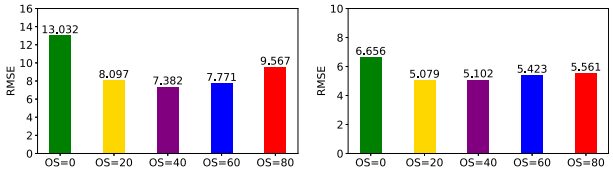

Fig. 8. Effects of oversampling between 0% and 80% for the Stevens Creek (left) and Vasona (right) reservoirs.

TABLE IV
IMPACT OF WAN AND GMM CLUSTERING INPUTS ON $RMSE$

Dataset	Type	no_WAN	no_PW	$PW_{factor} = 0.4$
Coyote	rolling	51.530	47.75	**45.373**
Coyote	3-day	472.578	435.032	**433.571**
Stevens Creek	rolling	9.051	9.470	**7.382**
Stevens Creek	3-day	80.089	**65.064**	71.303
Vasona	rolling	6.805	6.541	**5.137**
Vasona	3-day	19.792	19.032	**18.474**

oversampling ratio OS based solely on data statistics. Thus, we advise using a grid search strategy to identify the optimal values for these parameters, ensuring a more precise and data-driven approach to managing sample importance.

Fig. 8 shows the effects of oversampling between 0% and 80% for the Stevens Creek (left) and Vasona (right) reservoirs. For the Steven Creek reservoir, performance is enhanced by 23.7% compared to not oversampling when OS is set to 20%, i.e., 20% of the training data are oversampled. The optimal OS percentage for the Vasona reservoir is 40%, and its RMSE is lowered by 43.4%.

2) Effects of WAN and Segment-Wise Clustering: Considering the importance of point-wise clustering features, which capture the distributional characteristics of the input series, we evaluate the influence of segment-wise clustering (SW GMM) on overall model performance. To help understand this influence, Fig. 9 shows some randomly chosen examples of each SW GMM cluster when $M = 3$. It is easy to see that the clusters have samples with different shapes, as intended in our design. Furthermore, Table IV presents RMSE results for three reservoirs under different WAN configurations: (1) without WAN, (2) with segment-wise clustering only, and (3) with both point-wise and segment-wise clustering (optimal SW factor found via grid search).

The results show that SW factor enhances forecasting accuracy by allowing the model to capture segmented features over time. Compared to the no-WAN baseline, integrating SW alone improves 3-day prediction accuracy by 10.2% and rolling prediction accuracy by 2.2% on average. Further incorporating point-wise clustering yields rolling prediction RMSE reductions of 11.9%, 18.4%, and 24.5% for the respective reservoirs and 3-day RMSE reductions of 8.3%, 11.0%, and 6.7% respectively. These results show that combining segment-wise and point-wise clustering allows WAN to adaptively enhance forecasting accuracy across reservoirs.

3) Effects of Different Mixture Policies: As introduced in Section IV-E, in order to correct cluster imbalance, we merge the three sets of point-wise and segment-wise clustering feature series based on mixture weights learned from their respective GMM models. We arrange the M point-wise cluster series

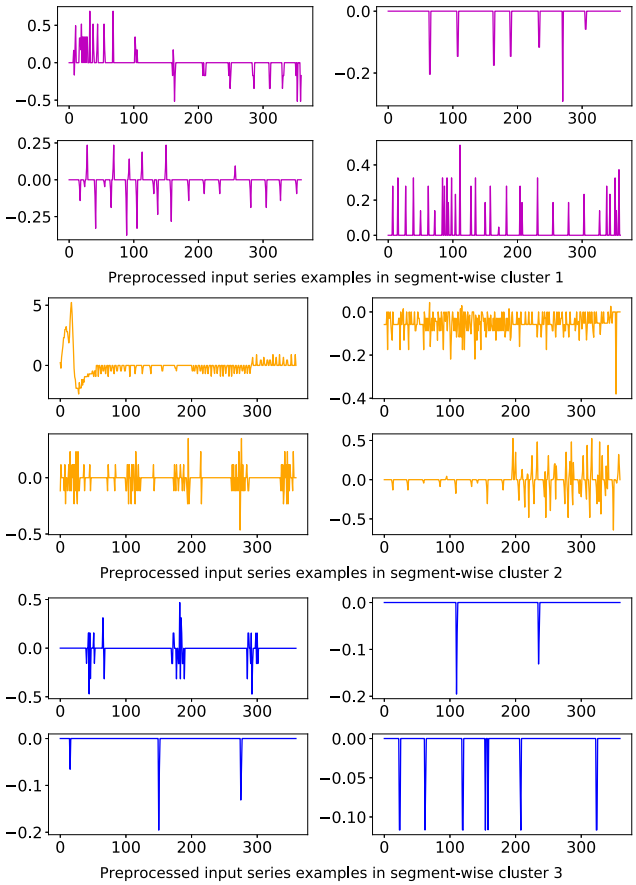

Fig. 9. Segment-wise clustering examples.

TABLE V
CLUSTER WEIGHTS BEFORE/AFTER MIXTURE

Coyote Reservoir	min	median	max
PW Cluster Weights	0.008	0.251	0.741
SW Cluster Weights	0.252	0.366	0.382
Mixture Cluster Weights	0.195	0.308	0.497
Stevens Creek Reservoir	**min**	**median**	**max**
PW Cluster Weights	0.017	0.269	0.714
SW Cluster Weights	0.122	0.329	0.549
Mixture Cluster Weights	0.283	0.299	0.418
Vasona Reservoir	**min**	**median**	**max**
PW Cluster Weights	0.009	0.160	0.831
SW Cluster Weights	0.209	0.366	0.425
Mixture Cluster Weights	0.217	0.263	0.520

in ascending order and the M segment-wise cluster series in descending order based on their GMM mixture weights. The imbalance of the clusters is relieved following the blending procedure, as Table V illustrates.

Additionally, we performed a full year rolling prediction on the Stevens Creek reservoir to demonstrate the significance of this policy, which can be seen in Fig. 10. We compare the worst policy, which involves merging the PW clusters in their initial order, with the best policy, which we just described. The

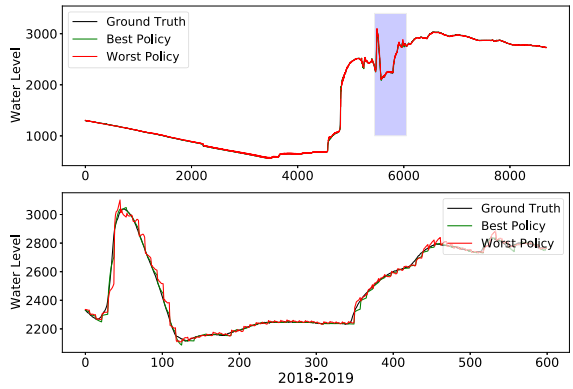

Fig. 10. Comparison of the effects of using the best and worst cluster mixture policies on a whole year rolling prediction of the Stevens Creek reservoir levels.

results show how the worst policy performs poorly as a result of significant imbalance.

VI. CONCLUSION

In this study, we introduce the Mixture Clustering Attention Neural Network (MC-ANN), a composite framework for univariate time series forecasting, tailored to effectively capture rare but critical extreme events in lengthy time series data. The MC-ANN model employs an innovative clustering-based sampling strategy to enhance the identification of crucial features, significantly improving prediction accuracy by learning from the data's mixed distribution. Demonstrated to excel in real-world applications, our model consistently updates short-term forecasts with high precision while also grasping long-term trends in extended three-day forecasts. Extensive testing on five real-world datasets has shown that MC-ANN outperforms existing methods, achieving a 10%–45% reduction in root mean square error for time series with high variance and extreme events. Additionally, the model's unique attention-based component weighting, which combines point-wise and segment-wise clustering, has been proven in ablation studies to further boost accuracy by over 20%.

ACKNOWLEDGMENT

This research was made possible through hardware resources provided by Supermicro and NVIDIA.

REFERENCES

[1] P. Hewage, M. Trovati, E. Pereira, and A. Behera, "Deep learning-based effective fine-grained weather forecasting model," *Pattern Anal. Appl.*, vol. 24, no. 1, pp. 343–366, 2021.
[2] S. Mohan, S. Mullapudi, S. Sammeta, P. Vijayvergia, and D. C. Anastasiu, "Stock price prediction using news sentiment analysis," in *Proc. IEEE 4th Int. Conf. Big Data Comput. Service Appl.*, 2019, pp. 205–208.
[3] Y. Chen et al., "MARLP: Time-series forecasting control for agricultural managed aquifer recharge," in *Proc. 30th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, New York, NY, USA, 2024, pp. 4862–4872, doi: 10.1145/3637528.3671533.
[4] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting," 2019, *arXiv: 1905.10437*.

[5] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "DeepAR: Probabilistic forecasting with autoregressive recurrent networks," *Int. J. Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.
[6] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 22419–22430.
[7] C. M. Bishop, "Mixture density networks," Aston University, WorkingPaper, 1994. [Online]. Available: https://research.aston.ac.uk/en/publications/mixture-density-networks
[8] Y. Pei, Y. Liu, N. Ling, L. Liu, and Y. Ren, "Class-specific neural network for video compressed sensing," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2021, pp. 1–5.
[9] D. Klotz et al., "Uncertainty estimation with deep learning for rainfall–runoff modeling," *Hydrol. Earth Syst. Sci.*, vol. 26, no. 6, pp. 1673–1693, 2022.
[10] Y. Li, J. Xu, and D. C. Anastasiu, "An extreme-adaptive time series prediction model based on probability-enhanced LSTM neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 8684–8691.
[11] G. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. San Francisco, CA, USA: Holden-Day, 1976.
[12] Z.-Y. Wang, J. Qiu, and F.-F. Li, "Hybrid models combining EMD/EEMD and ARIMA for long-term streamflow forecasting," *Water*, vol. 10, no. 7, 2018. [Online]. Available: https://www.mdpi.com/2073-4441/10/7/853
[13] J. Han, X.-P. Zhang, and F. Wang, "Gaussian process regression stochastic volatility model for financial time series," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 6, pp. 1015–1028, Sep. 2016.
[14] Q. Huang, H. Zhang, J. Chen, and M. He, "Quantile regression models and their applications: A review," *J. Biometrics Biostatistics*, vol. 8, no. 3, pp. 1–6, 2017.
[15] S. J. Taylor and B. Letham, "Forecasting at scale," *Amer. Statistician*, vol. 72, no. 1, pp. 37–45, 2018, doi: 10.1080/00031305.2017.1380080.
[16] A. Singh, R. K. Ranjan, and A. Tiwari, "Credit card fraud detection under extreme imbalanced data: A comparative study of data-level algorithms," *J. Exp. Theor. Artif. Intell.*, vol. 34, no. 4, pp. 571–598, 2022.
[17] R. Sen, H.-F. Yu, and I. S. Dhillon, "Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, Art. no. 435.
[18] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*. Hoboken, NJ, USA: Wiley, 2015.
[19] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "TimesNet: Temporal 2D-variation modeling for general time series analysis," 2022, *arXiv:2210.02186*.
[20] N. Kitaev, Ł. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," 2020, *arXiv: 2001.04451*.
[21] H. Zhou et al., "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 11106–11115.
[22] A. Das, W. Kong, A. Leach, R. Sen, and R. Yu, "Long-term forecasting with tide: Time-series dense encoder," 2023, *arXiv:2304.08424*.
[23] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting," 2022, *arXiv:2205.13504*.
[24] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," 2022, *arXiv:2211.14730*.
[25] Y. Liu et al., "iTransformer: Inverted transformers are effective for time series forecasting," 2023, *arXiv:2310.06625*.
[26] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture IE*, vol. 2, no. 1, pp. 1–18, 2015.
[27] N. Laptev, J. Yosinski, L. E. Li, and S. Smyl, "Time-series extreme event forecasting with neural networks at uber," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1–5.
[28] M. Zhang, D. Ding, X. Pan, and M. Yang, "Enhancing time series predictors with generalized extreme value loss," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 2, pp. 1473–1487, Feb. 2023.
[29] Z. Xiu, C. Tao, M. Gao, C. Davis, B. A. Goldstein, and R. Henao, "Variational disentanglement for rare event modeling," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 10469–10477.
[30] Y. Zhang et al., "Data regression framework for time series data with extreme events," in *Proc. IEEE Int. Conf. Big Data*, 2021, pp. 5327–5336.
[31] N. E. Day, "Estimating the components of a mixture of normal distributions," *Biometrika*, vol. 56, no. 3, pp. 463–474, Dec. 1969, doi: 10.1093/biomet/56.3.463.
[32] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2018, pp. 95–104.
[33] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "A comparison of ARIMA and LSTM in forecasting time series," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl.*, 2018, pp. 1394–1401.
[34] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 753–763.
[35] D. Cao et al., "Spectral temporal graph neural network for multivariate time-series forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 17766–17778.
[36] G. Spadon, S. Hong, B. Brandoli, S. Matwin, J. F. Rodrigues-Jr, and J. Sun, "Pay attention to evolution: Time series forecasting with deep graph-evolution learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5368–5384, Sep. 2022.
[37] N. Du and X. Liang, "Short-term water level prediction of Hongze lake by prophet-LSTM combined model based on LAE," in *Proc. 7th Int. Conf. Hydraulic Civil Eng. Smart Water Conservancy Intell. Disaster Reduction Forum*, 2021, pp. 255–259.
[38] Y. Le, C. Chen, T. Hang, and Y. Hu, "A stream prediction model based on attention-LSTM," *Earth Sci. Inform.*, vol. 14, pp. 1–11, Jun. 2021.
[39] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.
[40] Z. Chen, H. Yu, Y.-A. Geng, Q. Li, and Y. Zhang, "EvaNet: An extreme value attention network for long-term air quality prediction," in *Proc. IEEE Int. Conf. Big Data*, 2020, pp. 4545–4552.
[41] S. C. Ibañez, C. V. G. Dajac, M. P. Liponhay, E. F. T. Legara, J. M. H. Esteban, and C. P. Monterola, "Forecasting reservoir water levels using deep neural networks: A case study of angat dam in the Philippines," *Water*, vol. 14, no. 1, 2022. [Online]. Available: https://www.mdpi.com/2073-4441/14/1/34
[42] B. S. Chissom, "Interpretation of the kurtosis statistic," *Amer. Statistician*, vol. 24, no. 4, pp. 19–22, 1970.
[43] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.
[44] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2022, pp. 27268–27286.

Yanhong Li (Graduate Student Member, IEEE) is currently working toward the PhD degree in computer science and engineering with Santa Clara University's School of Engineering. Her research interests are centered on the development of advanced computational models and algorithms, with a keen focus on machine learning, pattern recognition, and deep learning technologies. She has a particular interest in time series representation learning, object detection, and tracking.

David C. Anastasiu (Member, IEEE) is an associate professor with the Department of Computer Science and Engineering, Santa Clara University. His research interests fall broadly at the intersection of artificial intelligence/machine learning, data mining, computational genomics, and high performance computing. Much of his work has been focused on scalable and efficient methods for analyzing sparse data. He has developed serial and parallel methods for identifying near neighbors, characterizing how user behavior changes over time, analyzing traffic based on video sensors, and methods for personalized and collaborative presentation of Web search results, among others. In the biomedical domain, he has worked on methods for sensory-based prediction of Autism in children, searching related biochemical compounds, and designating the severity of kidney disease. He serves on the program committees and senior program committees of the most prominent IEEE and ACM data science-related conferences and his work, which is funded by the National Science Foundation and several industrial partners, has been published in many top-tier conferences and journals.