zSHiFT: A Siamese Hierarchical Transformer Network for Zero Shot Time Series Forecasting

Harrison Thayer
Department of Computer Science
California State University, Fullerton
Fullerton, California 92831, USA
harrisonthayer@csu.fullerton.edu

Anand Panangadan
Department of Computer Science
California State University, Fullerton
Fullerton, California 92831, USA
apanangadan@fullerton.edu

***Abstract*—Zero shot time series forecasting is the challenge of forecasting future values of a time dependent sequence without having access to any historical data from the target series during model training. This setting differs from the traditional domain of time series forecasting, where models are typically trained using large volumes of historical data, from the same distribution. Zero shot time series forecasting models are designed to generalize to unseen time series by leveraging their knowledge learned from other, similar series during training. This work proposes two architectures designed for zero shot time series forecasting: zSiFT and zSHiFT. Both architectures use transformer models arranged in a Siamese network configuration. The zSHiFT architecture differs from the zSiFT by the introduction of a hierarchical transformer component to the Siamese network. These architectures are evaluated on vehicular traffic data in California available from the Caltrans Performance Measurement System (PeMS). The models were trained with traffic flow data collected in one region of California and then are evaluated by forecasting traffic in other regions. Forecast accuracy was evaluated at different time horizons (4 to 48 hours). The zSiFT model achieves a Mean Absolute Error (MAE) that is 8.3% lower than the baseline LSTM with attention mechanism model. The zSiFT model achieves an MAE which is 6.6% lower than zSHiFT's MAE.**

***Index Terms*—Time Series Forecasting, Zero-shot Forecasting, Traffic Forecasting, LSTM, Attention Mechanism, Transfer Learning, Sequence Modeling, Support-Based Forecasting, Deep Learning for Time Series, Transformer Networks, Siamese Networks.**

I. Introduction

A time-series is a sequence of data points collected at successive points in time. Time-series data arise in a number of domains, especially in finance and environmental studies [1]. With the growth of Internet-of-Things (IoT) technology, sensors embedded in several different types of environments continuously produce a time-series of sensor measurements. One of the most commonly used applications of such data streams is forecasting future values given a sequence of past measurements. For instance, sensors embedded in roads can measure the flow of vehicular traffic over them. Time-series forecasting in this context is predicting the amount of traffic over the next few hours.

While statistical models such as Autoregressive Integrated Moving Average (ARIMA) have traditionally been used for forecasting, the prediction accuracy in many applications is higher with more complex models built using machine learning from historic data. In particular, recurrent neural networks have performed well on this task [2]. The disadvantage of such machine learning approaches is the need for large amounts of historical time-series data. Such large datasets are not available in several applications or in specific regions due to lack of sensor infrastructure for collecting data. For instance, rural roads or smaller roads will have fewer sensors and hence lesser historical traffic data is available.

In order to address this need for data, "few-shot" machine learning methods attempt to use *related* datasets for the bulk of the training, and then use the limited data from the actual region of interest to fine-tune the model. Thus, few-shot learning is an instance of transfer learning [3]. In the extreme case, "zero-shot" learning methods complete all their training on the related data set and an instance from the true domain is only considered during prediction.

Several few-shot learning methods have been developed for classification. These approaches often rely on a siamese neural network architecture [4]. For classification, the key idea is to learn a difference vector between inputs instead of the true class label. This classification approach is not directly applicable to forecasting. Few-shot learning methods for time-series forecasting therefore rely on other methods, such as using a recurrent neural network with an attention mechanism [5].

In this work, we integrate these two approaches – combining recurrent neural network architectures with attention mechanism and the use of difference vectors. We propose two related architectures for zero-shot time-series forecasting: zSiFT and zSHiFT. Both architectures use transformer models arranged in a Siamese network configuration. The zSHiFT architecture differs from the zSiFT by the introducing a hierarchical transformer component to the Siamese network.

The zero-shot setting is applicable in real world domains where collecting historical data is difficult. For instance, zero-shot forecasting methods can enable traffic prediction along new roads, or along pedestrian pathways where sensors designed for vehicles are not useful for detecting pedestrian traffic. Zero-shot models enable forecasting without needing to be re-trained.

We evaluated these architectures by forecasting vehicular traffic traffic using data collected by the California Department of Transportation's (Caltrans) Performance Measurement System Data Source (PeMS). we use this dataset and traffic prediction application for an extensive evaluation of the architectures. The models were trained with traffic flow data collected in one part of California and then evaluated by forecasting traffic at different regions at varying time horizons (0 to 48 hours).

Our contributions are as follows: (1) We introduce and evaluate four neural network architectures for zero shot time series forecasting: An LSTM model with Attention, a Difference LSTM with Attention (DLA), a Siamese Transformer (zSiFT), and a Siamese Hierarchical Transformer (zSHiFT); (2) We propose the use of element-wise difference vectors combined with a support series to guide forecasts in the absence of a target series history; and (3) We evaluate each model on a real-world dataset, showing that zSiFT which uses a single resolution transformer encoder outperforms the more complex zSHiFT across most forecasting horizons.

II. Related Work

Recently extensive work has been done to improve the performance of zero shot time series forecasting models. For example a recurrent neural network (RNN) based model proposed in [6] addresses both zero shot and few shot forecasting by learning a shared latent embedding across multiple quantized time series. Experimental results on benchmark datasets show that this approach consistently outperforms Gaussian Processes and AR-based models in the zero shot setting. In parallel, efforts have been made to evaluated and enhance the capabilities of zero-shot time series foundations models (FMs). However, Toner et al. [7] report that FMs fail to generalize on certain types of data. Zeng et al. [8] critically examine the use of transformer based architectures for long term time series forecasting (LTSF). Their results reveal that despite their popularity, transformer models often inadequately capture temporal dependencies due to the permutation invariant nature of self attention, even when using positional encodings. Gruver el at. investigated the use of using Large Language Models (LLMs) for zero shot time series forecasting [9]. Their findings indicate that while LLMs are capable of performing zero shot time series forecasting, they suffer from limitations including as limited context windows. Merrill et al. investigated LLMs abilities to reason about time series in the zero shot setting, finding that highly capable models still struggle to answer questions about time series [10]. Furthermore the high inference cost of LLMs make them more difficult for use in time series forecasting [11].

Complementary to these works, Tran and Panangadan [12] proposed a few-shot time series forecasting approach based on Siamese neural networks with LSTM units. Their model learns to predict a difference vector between time series pairs rather than directly forecasting future values, thus allowing the models to generalize to unseen time series with only a few reference instances. This method also does not require retraining when exposed to new data types, making it useful for low-data or dynamic domains such as traffic forecasting. Our current work improves on this model by using transformer components in the architecture, and enabling zero-shot (as compared to few-shot) learning.

III. Approach

The problem of zero-shot forecasting can be stated as:

$$\hat{\mathbf{y}}_{t+1:t+H} = f_\theta(\mathbf{x}_{t-W+1:t}, \mathcal{S}), \tag{1}$$

where $\mathbf{x}_{t-W+1:t} \in \mathbb{R}^W$ is the input sequence, $\hat{\mathbf{y}}_{t+1:t+H} \in \mathbb{R}^H$ is the forecast, f_θ is the forecasting model, and $\mathcal{S}$ is an optional support set used in zero-shot settings.

We describe four zero shot time-series forecasting models. The first two models are meant to be used as baselines for performance comparison to our two new proposed architectures. Each model takes as input a time-series of length W, $\mathbf{x}_{t-W+1:t}$ (also called the query series) and produces a forecast of length H, $\hat{\mathbf{y}}_{t+1:t+H}$. H is the length of the forecast horizon. The models are trained on a dataset Z of L instances, $\mathbf{z}^{(j)}_{t-W+1:t}, j = 1, 2, \ldots, L$. Note that the training instances ($\mathbf{z}^{(j)}$) are drawn from a different distribution from the query series ($\mathbf{x}$). Thus, the forecast $\mathbf{z}$ is a zero-shot prediction of the input query. The above formulation describes univariate time-series and our evaluation also uses univariate time-series dataset. However, the architectures can be modified for multivariate time-series in a straightforward manner.

A. LSTM with Attention

This baseline model is a Long short-term memory (LSTM) architecture with an attention mechanism. LSTMs are a type of recurrent neural network (RNN) that has been shown to work particularly well with sequence data [13]. The Attention mechanism modifies the LSTM architecture by allowing the model to build its own internal varying length representation, enabling it to focus on only the most relevant parts of the input sequence when generating the output. The attention mechanism allows the model to understand and dynamically weigh the importance of different time steps in the input sequence, enabling it to capture patterns and dependencies. During forecasting, this model relies solely on the temporal dependencies learned through its recurrent structure and attention mechanism,i.e., it does not use training data directly during output generation.

B. Difference LSTM with Attention

The *Difference LSTM with Attention (DLA)* is an enhanced version of the LSTM with Attention model designed for zero shot forecasting and is employed as a baseline model. The DLA incorporates two additional components: a *support series* and a *difference vector*. This approach sets aside a randomly sampled proportion of the training dataset Z, called the *Support dataset (SV, $\mathcal{S} \subset Z$)*, for use during forecasting. The support series is the one instance in the

Support dataset that is most similar to the query series. In this work, we quantify similarity by selecting the instance with the smallest squared Euclidean distance from the query series. Specifically, the support series of input x, $S(x)$, is defined as:

$$S(x) = z^*, d(x, z^*) = \min_{z \in \mathcal{S}} d(x, z)$$

$$d(x, z) = \sum_{i=1}^{m} (x_i - z_i)^2$$

The function of the support series is to provide the model with an approximation for what an accurate forecast will resemble. In order to accommodate for the fact that the query and support vectors are drawn from different distributions, the model also incorporates a difference vector. The difference vector is the vector of element-wise differences between the support and query series. Specifically, the difference vector of input x, $DV(x)$, is defined as

$$DV(x) = x - S(x)$$

By explicitly modeling both the similarity and the differences, the DLA model enhances its ability to accurately forecast the query series.

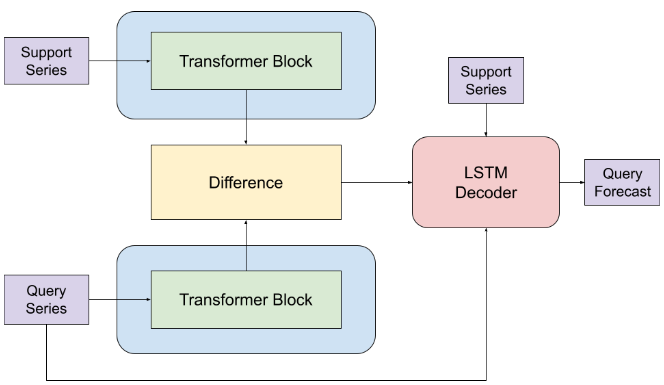

Fig. 1: The proposed zSiFT architecture

C. Siamese Transformer

The Siamese Transformer for zero shot time series forecasting architecture (zSiFT) uses two transformers arranged in a Siamese network configuration to capture complex temporal dependencies through their multi-head self-attention mechanism. This model also incorporates the above-defined support vector. The Siamese network is composed of twin transformer models, where model 1 takes as input the data points from the query series, and model 2 takes as input data points from the support series. The Siamese-twin network creates an encoding of each series, and then calculates the element-wise difference of the two vectors. Finally, the query series, the support series, and the element-wise difference of the two transformer encodings are used as inputs into the LSTM decoder, which generates the final forecast. The zSiFt architecture is shown in Figure 1.

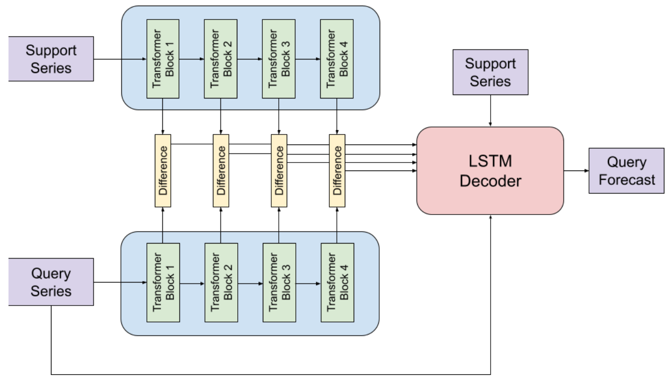

Fig. 2: The proposed zSHiFT architecture

D. Siamese Hierarchical Transformer

The Siamese hierarchical transformer network for zero shot time series forecasting (zSHiFT) builds upon the zSiFT architecture by introducing a hierarchical transformer component to the Siamese network. Instead of taking the element-wise difference of two identical transformers from the Siamese network, SHiFT uses multiple pairs of identical transformers. Each pair encodes information at a different temporal resolution. Finally, the query series, the support series, and the vector of element-wise difference at each resolution are used as inputs into the LSTM decoder, which generates the final forecast. The zSHiFT architecture is shown in Figure 2.

Although Transformers generally excel at capturing complex dependencies in sequential data, our initial experiments indicated that employing Transformers directly as decoders resulted in inferior performance compared to using LSTMs. Therefore, we opted to utilize LSTMs for the final decoding step in the zSiFT and zSHiFT architectures rather than Transformers. All four models are trained using Mean Absolute Error (MAE) as the loss function, which is commonly used in time series tasks due to its robustness to outliers and interpretability [14].

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \,, \tag{2}$$

where y_i is the true value, $\hat{y}_i$ is the predicted value, and n is the number of forecasted time steps. Minimizing the MAE encourages the models to produce forecasts that more closely align with the ground truth values while treating all errors equally, regardless of magnitude.

IV. Datasets

1) Dataset Creation: Our training, validation, and testing datasets are created from California Department of Transportation (Caltrans) Performance Measurement System Data Source (PeMS) [15]. The Caltrans data is collected from nearly 40,000 individual detectors across the state of California. The data is collected in 5 minute intervals and aggregated into the Data Clearinghouse. We used this data to train, validate, and test each forecasting model.

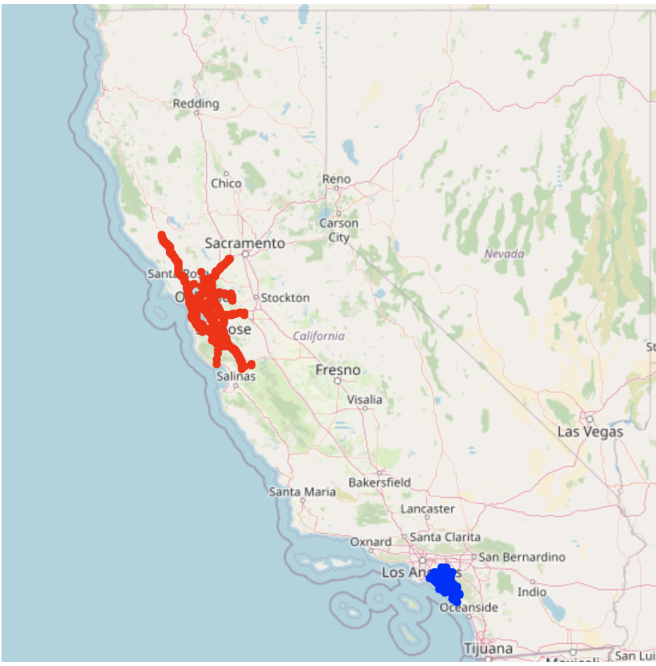

Fig. 3: Map of California showing station locations: red dots indicate test stations (District 4) and blue dots indicate training stations (District 12).

2) Data Pre-Processing: We downloaded data from the PeMS Data Clearinghouse for training and evaluation. The Data Clearinghouse datasets contain a variety of features collected at a 5-minute sampling interval for each station. For our work, we selectively retained only the timestamp, station identifier (a unique identifier for each station), and total flow. During this stage, we discard all other features. To reduce data noise and enhance forecasting reliability, we down sampled the data from its original 5-minute intervals into 30 minute intervals. This aggregation was performed by summing the traffic flows recorded within each 30-minute interval, therefore reducing the dataset's resolution. Our training dataset consists of data from Orange County, California (Caltrans District 12), covering the full year of 2023. The testing dataset includes data from the Bay Area and Oakland (Caltrans District 4) for the first three months of 2023. This intentional geographic separation allows us to evaluate the models' ability to perform under zero-shot conditions, assessing their generalization to locations not seen during training.

Furthermore, the training and testing datasets are comprised solely of stations which are a part of a Mainline as determined by Caltrans. Stations with incomplete or missing data were identified and removed from the datasets to ensure data integrity and model reliability. Figure 3 shows the locations of the stations used for training and testing.

3) Dataset Windowing: The dataset consists of four-day time series, each containing 192 data points. Each data point represents the flow at a given station over a 30 minute interval. The data points within each four-day period must form a continuous, uninterrupted sequence. We utilize the sliding window method to create the four-day time series, ensuring that the windows are each 4 continuous days starting at midnight on the first day and ending at midnight on the fifth day. We opt for a 30 minute traffic flow forecast to reduce the amount of noise in the dataset.

4) Dataset Partitioning: We create two datasets: one containing all of the data from District 12 in 2023, and the other comprising data from the first 3 months of 2023 for District 4. The District 12 dataset is further partitioned into training and validation datasets based on station IDs utilizing a 70/30 split. Each of the three resulting datasets – training, validation, and testing – are then further subdivided into query and support sets using an 85/15 split.

V. Results and Discussion

Our models are trained using the National Research Platform (NRP) Nautilus Hypercluster. Training was performed using an Intel(R) Xeon(R) Gold 6230 CPU, and an NVIDIA GeForce RTX 2080 Ti. All four models were implemented using PyTorch v2.2.1. Other libraries used in this work include Pandas v2.2.3, NumPy v1.26.3, Scikit-Learn v1.6.1, Python 3.10.13, Selenium v4.25.0.

We evaluated the performance outcomes of the forecasting models using standard regression metrics: Mean Absolute Error (MAE), Mean Square Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2). Each of these metrics are calculated using the previously described test dataset. Table I provides a comparison of each forecasting model's performance metrics.

TABLE I: Forecasting Model Performance Comparison

Model	MAE	MSE	RMSE	MAPE (%)	R^2
LSTM*	144.89	48151.62	219.43	17.42	0.9482
DLA	146.89	50632.10	225.01	17.51	0.9454
zSiFT	**132.88**	**42630.77**	**206.47**	**14.74**	**0.9533**
zSHiFT	142.31	47723.11	218.45	17.10	0.9486

* LSTM with Attention mechanism.
MAE and RMSE are measured in number of vehicles per 30 minutes.

The baseline LSTM with Attention model performs well with an MAE of 144.89 and a relatively high R^2 of 0.9482. However, this model does not leverage any external context or support series, relying solely on the input series. Despite this limitation, its performance provides a strong benchmark.

The Difference LSTM with Attention (DLA) model uses additional inputs of the support and difference vectors. The intended effect of these additional inputs is to improve the model's ability to generalize to new inputs. However, its performance metrics such as MAE of 146.89, and MSE of 50632.10 suggest that this model is not able to take advantage of the additional support and difference vectors.

The zSiFT model significantly outperforms both the LSTM with Attention and the Difference LSTM with Attention. Specifically, the zSiFT model achieves an MAE of 132.88, as opposed to 144.89 for the LSTM with attention, marking

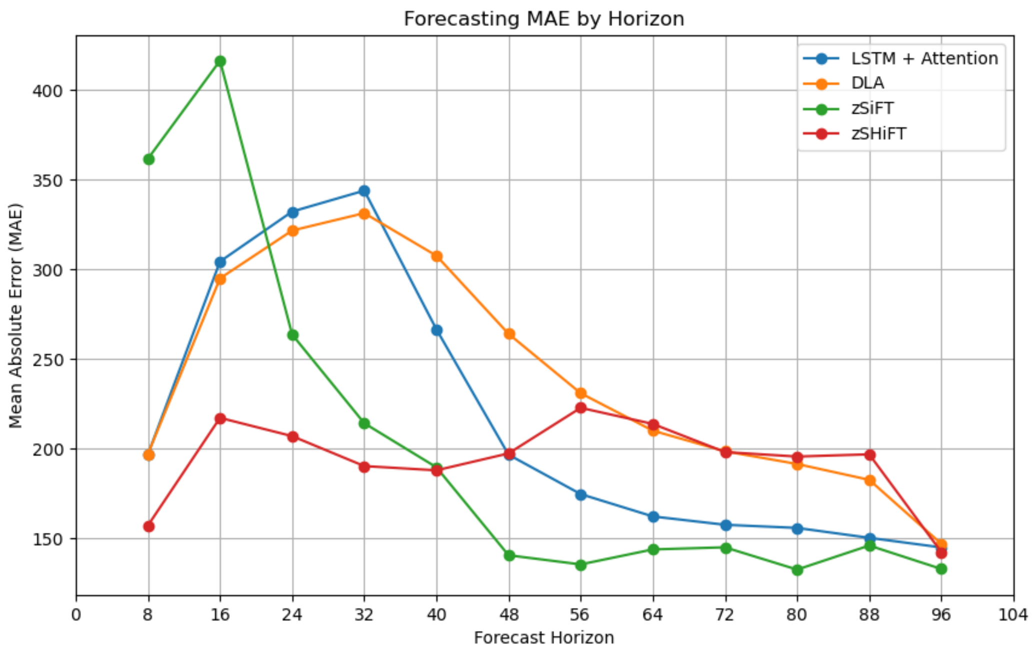

Fig. 4: Mean Absolute Error (MAE) [Vehicles] across different forecasting horizons for all evaluated models. On the x-axis lies the forecasting horizon, whereas the y-axis indicates the corresponding MAE values.

an 8.3% reduction in MAE. This model effectively highlights the benefit of utilizing a Siamese Transformer architecture to encode and compare query and support vector relationships. Through the combination of the transformers self-attention mechanism, and the explicit modeling of the difference vector, the zSiFT model is able to better capture the temporal dependencies in the data, and better generalize patterns across the regions and stations in the dataset.

The zSHiFT model, which includes a hierarchical multi-resolution component in its Siamese network, performs worse than the zSiFT model. We used 4 pairs of transformers as the hierarchical component in our evaluation. In our testing, the zSiFT model achieves an MAE of 132.88, which is 6.6% lower than zSHiFT's MAE of 142.31. While zSHiFT extends the capabilities of zSiFT to allow the model to capture temporal dependencies at multiple resolutions, this added complexity likely introduced overfitting. Nonetheless, zSHiFT still slightly outperforms the baseline LSTM with Attention and DLA models.

Figure 4 displays the MAE across the different tested forecasting horizons for all four models. The forecast error of all models initially increase with horizon length but then show a general downward trend in MAE as the forecasting horizon further increases; this reflects the strong daily seasonal component in the dataset. Before the horizon=48 data point mark (one day), all the models show relatively high forecast error. This indicated that short-term forecasts of vehicular traffic is particularly sensitive to random fluctuations in the

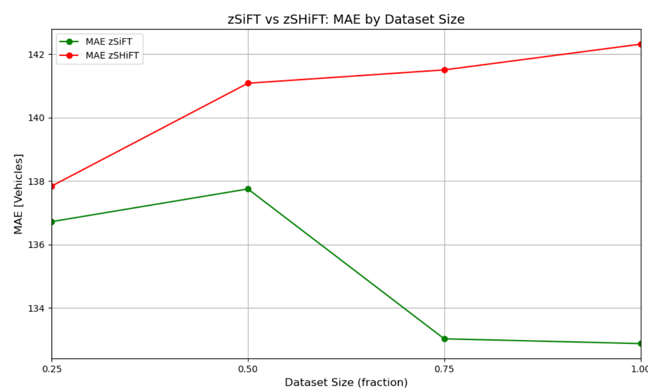

Fig. 5: Performance (MAE) and for zSiFT and zSHiFT as a function of training dataset size.

series. While zSHiFT performs better than all other models at horizon length less than 40, zSiFT retains the lowest MAE after this horizon further reinforcing the advantage of its transformer-based Siamese architecture. These results indicate that zSHiFT may perform better than zSiFT and all other models for short horizon forecasting.

Figure 5 displays test MAE of the zSiFT and zSHiFT models as the size of the dataset increases. The zSiFT model demonstrates improved generalization performance as the training set size increases, with the test MAE decreasing from 136.72 at 25% of the data to 132.88 at 100%. This suggests that zSiFT is able to utilize larger training sets

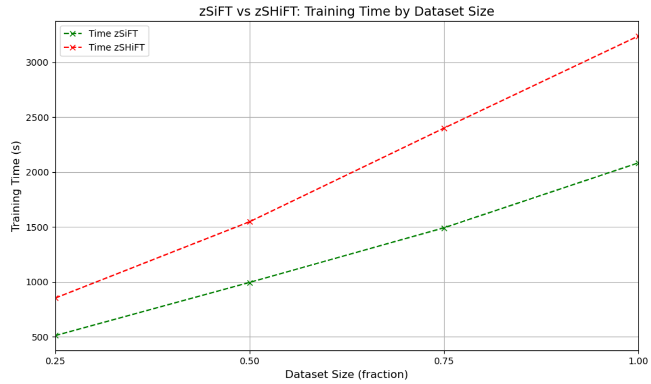

Fig. 6: Training time (seconds) for zSiFT and zSHiFT as a function of training dataset size.

to refine its temporal representations and produce more accurate forecasts. In contrast, zSHiFT exhibits a consistent degradation in its test performance as the size of the training set increases. As the dataset grows from 25% to 100%, the test MAE of the zSHiFT model increases from 137.84 to 142.31. This decrease in performance suggests that the increased complexity introduced by zSHiFT's hierarchical Siamese architecture may lead to overfitting that becomes more prominent at larger input data size.

Figure 6 highlights the training time of the zSiFT and zSHiFT models as the size of the dataset increases. As expected, the training time increases linearly with the size of the training dataset. Note that zSHiFt requires substantially more training time compared to zSiFT at all training set sizes.

VI. Conclusions

This work described two architectures designed for zero shot time series forecasting: zSiFT and zSHiFT. Both architectures used transformer models arranged in a Siamese network configuration. These architectures were evaluated on vehicular traffic data in California collected from the Caltrans Performance Measurement System (PeMS). Forecast accuracy was evaluated at different time horizons (4 to 48 hours) using multiple metrics, including MAE, MSE, MAPE, and the coefficient of determination (R^2). These models were compared with baseline LSTM models with attention mechanism, modified to take advantage of a support set for zero-shot forecasting. The zSiFT model achieves an MAE that is 8.3% lower than the baseline LSTM with attention mechanism model. This shows that the Siamese configuration of the transformer components is required to take advantage of the support dataset for zero shot forecasting. The zSiFT model achieves an MAE which is 6.6% lower than zSHiFT's MAE. The added complexity of zSHiFT likely introduced overfitting. Nonetheless, zSHiFT still slightly outperforms the baseline LSTM with Attention models.

This work focused on zero-shot time series forecasting. One immediate direction for future work is to extend both the zSiFT and zSHiFT architectures to handle multivariate time series. This could allow these models to capture interdependence between multiple signals. Incorporating multiple signals could improve the results of the models by providing a richer temporal context and improve the generalization performance on unseen data. Another direction for future work involves improving the mechanism by which the models select similar time series from the support set. In this work, the similarity between time series was calculated using Euclidean Distance. We propose to incorporate Dynamic Time Warping (DTW), in order to allow the models to select better support series.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. 2125654.

References

[1] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International journal of forecasting*, vol. 22, no. 3, pp. 443–473, 2006.
[2] Y. Liu, C. Gong, L. Yang, and Y. Chen, "Dstp-rnn: A dual-stage two-phase attention-based recurrent neural network for long-term and multivariate time series prediction," *Expert Systems with Applications*, vol. 143, p. 113082, 2020.
[3] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "Meta-learning framework with applications to zero-shot time-series forecasting," 2020. [Online]. Available: https://arxiv.org/abs/2002.02887
[4] G. Koch, R. Zemel, R. Salakhutdinov *et al.*, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2, no. 1. Lille, 2015, pp. 1–30.
[5] T. Iwata and A. Kumagai, "Few-shot learning for time-series forecasting," 2020. [Online]. Available: https://arxiv.org/abs/2009.14379
[6] B. P. Orozco and S. J. Roberts, "Zero-shot and few-shot time series forecasting with ordinal regression recurrent neural networks," 2020. [Online]. Available: https://arxiv.org/abs/2003.12162
[7] W. Toner, T. L. Lee, A. Joosen, R. Singh, and M. Asenov, "Performance of zero-shot time series foundation models on cloud data," 2025. [Online]. Available: https://arxiv.org/abs/2502.12944
[8] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" 2022. [Online]. Available: https://arxiv.org/abs/2205.13504
[9] N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson, "Large language models are zero-shot time series forecasters," *Advances in Neural Information Processing Systems*, vol. 36, pp. 19 622–19 635, 2023.
[10] M. A. Merrill, M. Tan, V. Gupta, T. Hartvigsen, and T. Althoff, "Language models still struggle to zero-shot reason about time series," 2024. [Online]. Available: https://arxiv.org/abs/2404.11757
[11] S. Samsi, D. Zhao, J. McDonald, B. Li, A. Michaleas, M. Jones, W. Bergeron, J. Kepner, D. Tiwari, and V. Gadepally, "From words to watts: Benchmarking the energy costs of large language model inference," in *2023 IEEE High Performance Extreme Computing Conference (HPEC)*, 2023, pp. 1–9.
[12] V. Tran and A. Panangadan, "Few-shot time-series forecasting with application for vehicular traffic flow," in *2022 IEEE 23rd International Conference on Information Reuse and Integration for Data Science (IRI)*, 2022, pp. 20–26.
[13] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The performance of lstm and bilstm in forecasting time series," in *2019 IEEE International conference on big data (Big Data)*. IEEE, 2019, pp. 3285–3292.
[14] N. Reich, J. Lessler, K. Sakrejda, S. Lauer, S. Iamsirithaworn, and D. Cummings, "Case study in evaluating time series prediction models using the relative mean absolute error," *The American Statistician*, vol. 70, 02 2016.
[15] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia, "Freeway performance measurement system: mining loop detector data," *Transportation Research Record*, vol. 1748, no. 1, pp. 96–102, 2001.