# CS-NET: STRUCTURAL APPROACH TO TIME-SERIES FORECASTING FOR HIGH-DIMENSIONAL FEATURE SPACE DATA WITH LIMITED OBSERVATIONS

*Weiyu Zong*[1], *Mingqian Feng*[2], *Griffin Heyrich*[1], *Peter Chin*[1,3]

[1]Boston University Department of Mathematics and Computer Science
[2] Johns Hopkins University Department of Applied Mathematics and Statistics
[3] Dartmouth College Thayer School of Engineering

## ABSTRACT

In recent years, deep-learning-based approaches have been introduced to solving time-series forecasting-related problems. These novel methods have demonstrated impressive performance in univariate and low-dimensional multivariate time-series forecasting tasks. However, when these novel methods are used to handle high-dimensional multivariate forecasting problems, their performance is highly restricted by a practical training time and a reasonable GPU memory configuration. In this paper, inspired by a change of basis in the Hilbert space, we propose a flexible data feature extraction technique that excels in high-dimensional multivariate forecasting tasks. Our approach was originally developed for the National Science Foundation (NSF) Algorithms for Threat Detection (ATD) 2022 Challenge. Implemented using the attention mechanism and Convolutional Neural Networks (CNN) architecture, our method demonstrates great performance and compatibility. Our models trained on the GDELT Dataset finished 1st and 2nd places in the ATD sprint series.

***Index Terms***— time-series forecasting, high-dimensional data, CNN, short-sequence, attention

## 1. INTRODUCTION

With data of all types becoming more and more abundant, real-time time series forecasting is taking on a more important role than ever in decision-making throughout many aspects of various domains, such as public health and safety[1], energy[2], transportation[3], and business[4]. Recently, due to the rapid rise in complexity and size of time series data, a wide variety of machine learning models have been explored in forecasting time series. The methods include, but are not limited to, Temporal Convolutional Neural Network (TCNN), Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), Elman Recurrent Neural Network (ERNN), and Multilayer Perceptron (MLP)[5]. These models generally work well with low dimensional data with a high number of observations yet still suffer from numerous problems[5]. Particularly, the performance of the models across different datasets has a large variance. Notably, current top performers like LSTM and GRU suffer from extremely high training and inference times compared to their counterparts. Existing CNN-based models like TCNN take the lead in their efficiency, but their performance drops drastically when handling data with high dimensionality. The trade-offs mentioned above lead us to rethink these approaches.

High-dimensional multivariate datasets undoubtedly pose challenges to existing forecasting models. However, they provide additional information that we can exploit. Unlike particles in an ideal gas, most events in the real world are not independent. Thus, we make the assumption that there exist underlying interactions between different events occurring at different locations and times. By this assumption, capturing the spatio-temporal relationships between the historical observations of the time series could help us interpret and forecast the states of future time frames, which are stochastic in nature. In this paper, we propose a novel strategy for feature extraction and neural network architecture design. Our strategy consists of three components–data pre-processing using structural decomposition, attention-based encoder, and convolution-based signal transform. Each component is laid out in detail in the Methodology Section.

The goal of this research work is aligned with the ATD-2022 Challenge and its sponsors, NSF and National Geospatial-Intelligence Agency (NGA). We aim to develop efficient algorithms that makes accurate predictions for high-dimensional time series data using limited historical observations which can be used in supporting public health and safety.

## 2. PROBLEM FORMULATION

The original problem that our model aimed to solve is proposed by the NSF ATD 2022 Challenge. Participants of the challenge are given a dataset derived from the GDELT project dataset. The GDELT project monitors print, broadcasts, and web media to record events across the globe and attribute them to state actors using the Conflict and Mediation Event Observations (CAMEO) coding system. In the CAMEO coding system, events are categorized into 20 distinct types according to their severity and rarity, ranging from "Making

Public Statements" to "Engage In Unconventional Mass Violence." For example, on one side of the spectrum, events like "Making Public Statements" could be happening at every moment around the world, which means that we have abundantly available past observations for these types of events. However, rare and catastrophic events, such as "Engage In Unconventional Mass Violence," would have very limited available past observations. Organizers of ATD 2022 processed the raw GDELT data into a weekly-aggregated, geographic-region-level view.

The resulting dataset consists of the counts of the 20 CAMEO event types for 260 geographic regions across a 215-week window, which is a 215 by 5,200 table. The task is to use available past observations to forecast the next four states of the world in the future 4-week horizon. Namely, time series observations $T_{0,\ldots,k}$ are given, where each $T_n \in N^{5,200}$ represents the state of the world at time $n$. Participants must produce a forecast function

$$f : T_{0,\ldots,k} \rightarrow T_{k+1,\ldots,k+4}$$

where $T_{k+1,\ldots,k+4}$ are the next four vectors representing future states of the world.

The backtesting procedure of ATD 2022 uses an expanding window starting with 100 weeks of past observations as the training set. The window expands for one observation at a time until the window exhausts the entire dataset. The consecutive 4 weeks of observations following the expanding window are then used as the testing set to evaluate the model performance.

## 3. METHODOLOGY

### 3.1. Data Preprocessing: Structural Decomposition

In our context, it is safe to assume that a sequence of numbers, such as a time series dataset, often represents interpretable measurements from the real world. Given this assumption, we propose that for any high-dimensional longitudinal dataset, we consider the key structural dimensions and arrange the data set accordingly, which can help deep-learning models converge faster. For example, we can decompose the ATD version of the GDELT Dataset into three major dimensions–Time, Event types, and Region. As illustrated in figure 1, we obtained a cuboid-shaped data block using the proposed approach, where each dimension corresponds to time, event categories, and space, respectively. In other words, every cross-section of the cuboid in the spatial dimension is a separate panel data of time and event types.

### 3.2. Attention-based encoder

In many situations, we do not necessarily have prior knowledge about the relationships between different time series in our dataset. Inspired by the success of attention-based structures in natural language processing, we included the attention-based encoder in our model. We aim to use this encoding layer as an additional effort to capture potential connections across the different time series. In our case, as shown in Figure 2, we applied an attention-based encoder throughout the entire spatial dimension to emphasize underlying associations across different geographic locations. The encoder layer is the same as the structure proposed in the original paper [6].

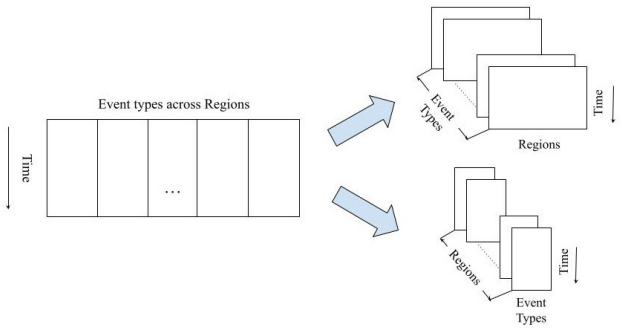


**Figure 1**: The proposed method for data preprocessing–Structural Decomposition. The input dataset is decomposed into two cuboid-shaped data blocks consisting of layers of panel data. In the ATD Dataset the panels are "Time" vs. "Regions" and "Time" vs. "Event types." Note that different datasets can be decomposed differently according to the Key Dimensions.

### 3.3. Convolution-based signal transform

Signal transform techniques like the Fourier transform are very commonly used in signal processing. It is capable of revealing essential characteristics of a signal by approximating the signal as a linear combination of its basis frequencies. Previous research suggests that the use of a windowed Fourier transform enables a better interpretation of the randomness in a given signal [7]. Notice that when the input signal is projected into an arbitrary Hilbert space, the Fourier transform operation can be interpreted as a change of basis. For example, suppose that the input signal F is in time domain t; we can write the following to express a windowed Fourier transform,

$$F = \langle \vec{f}, e^{i\omega t} \rangle = \int_{-\infty}^{\infty} f(t) g(t-s) e^{i\omega t}\, dt$$

where $f$ is the input signal, $g$ is a window function, and $e^{i\omega t}$ is the family of trigonometric functions serving as the orthonormal basis in the Hilbert space.

To expand on this idea, we need to produce a transformation that can decompose our panel of time series into a set of basis, which we can use as features to make combinations and generate predictions. In our case, every cross-section of

the reshaped cuboid data block is a 2-D panel. For instance, if we take a slice of the spatial dimension, we will obtain a time versus event type panel. Inspired by the work in [8], image representation using 2-D Gabor Wavelets, we realized that applying the concept mentioned above on this slice of data can be implemented efficiently using a 2-D convolution. i.e.,

$$F_n(t,s) = \sum_x \sum_y f(x,y) k_n(t-x, s-y)$$

where $k_n \in K$ is $n^{\text{th}}$ kernel of the set of kernels (basis) to be learned in the training process. As shown in figure 2, these extracted features are fused through an MLP layer to generate predictions.

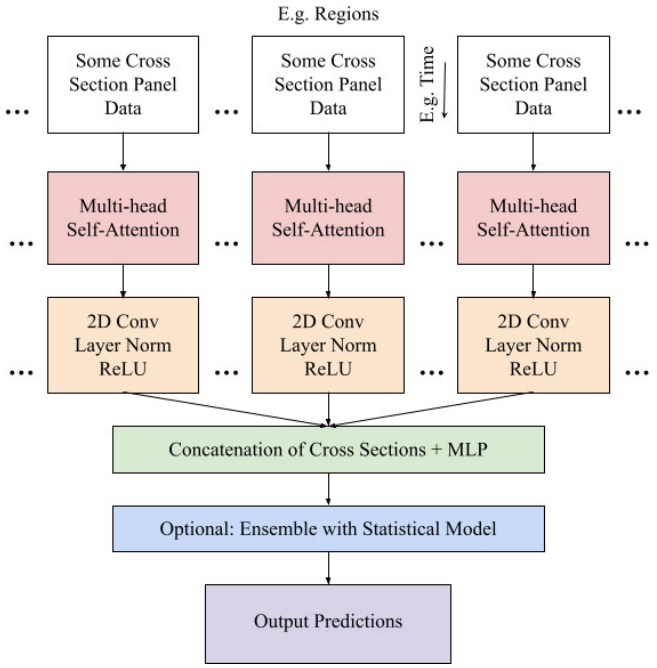


**Figure 2**: An overview of the proposed architectural design

### 3.4. Optional Model Ensemble Layer

During the model development process, we noticed that statistical models like ARIMA models tend to have a higher bias, making them better at following the general trend in the time series data. However, machine learning-based models are more likely to have a higher variance, especially when given limited observations as in our case. As a result, machine learning-based methods are better at predicting localized and drastic changes in the data. Depending on the specific dataset in consideration, we propose an optional layer that performs a weighted sum of our model predictions and additional statistical model predictions to balance the bias-variance tradeoff problems. In our case, we ensemble our model predictions with predictions from an additional Vector Auto-regression model (VAR).

### 3.5. Metrics

Since the ATD version of the GDELT Dataset contains a total of 5,200 time series with distinct scales and the values of some observations are zero, we need a metric invariant to the magnitude of data points and is capable of handling zero as truth values. Therefore, we chose Mean Absolute Scaled Error (MASE) as our primary metric. MASE is defined as the ratio between the mean absolute error of the forecast values and the mean absolute error of the in-sample one-step naive forecast [9]. i.e.,

$$MASE = \sum_{k=1}^{h} \frac{1}{h} \frac{|y_{t+k} - \hat{y}_{t+k}|}{(n-1)^{-1} \sum_{i=2}^{n} |y_i - y_{i-1}|}$$

where $h$ is the length of the forecasting horizon; $n$ is the total number of observations; $\hat{y}$ is the predicted value for observation $y$.

Besides the accuracy of the model, we also want to compare different model's capabilities in avoiding making large forecasting errors. Thus, we have also included Mean Squared Error (MSE) as a secondary reference since MSE, as defined in the expression below, emphasizes the large forecasting errors.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

where $n$ is the total number of observations; $\hat{y}$ is the predicted value for truth value $y$.

## 4. EXPERIMENTAL SETUP

### 4.1. Baselines for model comparisons

To assess the performance of our proposed method, we are using three state-of-the-art deep learning models and one classic statistical model for time series forecasting as baselines. Namely, Google AI's Temporal Fusion Transformer (TFT) [10], N-Beats Forecaster from Element AI [11], DeepAR Forecaster from Amazon Research [12], and Vector Autoregression (VAR) Model. The three deep learning models were implemented with GluonTS, a Python library [13], and the VAR model was implemented using Python Statsmodels library [14]. The parameters for each model are chosen carefully using grid search method around the default values recommended by the library implementations to achieve their best performance.

### 4.2. Dataset preparation and training details

The performance of our proposed model is demonstrated on two separate datasets. 1) ATD 2022 Dataset, 2) Wikipedia Web Traffic Dataset. Dataset 1 was given to participants for model development. The details of our proposed Structural

| | ATD Data | | Wikipedia Data | |
|---|---|---|---|---|
| Model Names | MASE | MSE | MASE | MSE |
| CS-Net 1 | 1.126 | 73.90 | 0.773 | 785.47 |
| CS-Net 2 | 1.032 | 61.08 | 0.731 | 736.19 |
| **CS-Net 3** | **1.015** | **60.50** | **0.694** | **718.19** |
| DeepAR | 1.136 | 74.04 | 0.955 | 964.64 |
| TFT | 1.078 | 63.43 | 0.752 | 748.01 |
| VAR | 1.193 | 68.19 | 0.977 | 973.69 |
| N-Beats | 1.112 | 73.43 | 0.748 | 746.44 |

**Table 1**. Performance of different models on two datasets using MASE and MSE as metrics

Decomposition procedure on Dataset 1 are already described in the previous sections, which we do not repeat here. Dataset 2 is obtained from a Kaggle competition [15] with the goal of predicting future web traffic of Wikipedia web pages. The raw dataset contains 145, 063 time series. We randomly selected a subset of 1,400 distinct time series of web pages for active keywords in multiple languages. Each of these is the recorded daily web traffic spanning 500 days, resulting in a 500 by 1,400 table as the input. In addition, the name of each time series also contains information regarding the language a web page is written in, the device type users used to access a web page, and keywords related to a web page. The key structural dimensions that we are considering here are Time, User Device types, Language, and Keyword categories.

For Dataset 1, all models are trained on a sliding window of 170 observations in size to predict the next 8 observations. Similarly, for Dataset 2, we trained the models on a sliding input window of size equal to 300 observations to predict the next 4 observations. Note that we choose different training window sizes because Dataset 2 has more number of observations. In addition, our model has already demonstrated strong performance in the ATD 2022 challenge with a forecast horizon equal to 4. So we decided to test our model's performance over a longer forecast horizon.

## 5. RESULTS AND ANALYSIS

### 5.1. Ablation and Comparison Analysis

We made three variations of our model to quantify the effectiveness of each of our proposed architectural components. For simplicity, we named our proposed model design Cross-Sectional-Net (CS-Net). CS-Net 1, 2, and 3 correspond to the three variations. CS-Net 1 only has the Convolution-based signal transform. Predictions from CS-Net 2 are a weighted sum of CS-Net 1 outputs and predictions made by a VAR forecaster. Lastly, CS-Net 3 has all the components proposed in the Methodology section, i.e., Convolution based signal transform, Attention-based encoder, and Model ensemble.

According to Table 1, we see that regardless of the dataset or metric used, the proposed Convolution-based signal transform across data sections alone yield a relatively competitive performance. Its metric scores is close to that of TFT and N-Beats forecasters. The inclusion of a statistical model ensembling further improved the proposed method's performance, which verified our hypothesis that variances and biases need to be balanced. It is worth noting that even though classical statistics models like VAR do not show great performance when used alone, they can be helpful in supporting the robustness of deep learning-based predictions.

We also noticed that in the ATD Dataset, the improvements from adding multi-head self-attention layer is not as significant as in Wikipedia Traffic Dataset. This is reasonable because ATD Dataset has less available data for training. We observed that with limited data, models may have over-fitted to part of time series and not have fully converged in others. To counter this issue, we used regularization techniques, such as dropout and adding regularization parameter in the Adam loss function. We also expect more notable improvements given relative more historical observations.

With all of the proposed components integrated, CS-Net 3 outperforms the other baseline models in both datasets. Particularly, it takes a strong lead in the MSE metric, indicating that it successfully avoided making large prediction errors.

### 5.2. Limitations

Despite that the proposed method outperforms state-out-the-art architectures like TFT, N-Beats and DeepAR Forecaster, the proposed method relies on a stronger assumption on a given dataset that it must contain some type of structure. This assumption reduces its generalizability on different datasets. In addition, TFT provides interpretability of variable importance, which could be useful in applications.

## 6. CONCLUSIONS

In this paper, we adopted a novel approach to handle high-dimensional multivariate time series forecasting tasks with limited availability of historical observations. The proposed approach has shown success in the NSF ATD 2022 Challenge. This method also outperforms other cutting-edge methods in time series forecasting in subsequent experiments on additional datasets, confirming the effectiveness of the proposed method. This research work can potentially be used in supporting fields of public health and safety.

## 7. ACKNOWLEDGEMENT

This work was funded by the United States National Science Foundation Division of Mathematical Sciences, under the award grant NSF-DMS 1737897. The authors would like to thank Penn State Applied Research Laboratory for organizing the NSF ATD 2022 Challenge. All authors contributed equally to this work. Listing order is random.

## 8. REFERENCES


[1] C. Vladescu V. Olsavszky, M. Dosius and J. Benecke, "Time series analysis and forecasting with automated machine learning on a national icd-10 database," *International journal of environmental research and public health*, vol. 17, 2020.

[2] Aditya Ashok, Manimaran Govindarasu, and Venkataramana Ajjarapu, "Online detection of stealthy false data injection attacks in power system state estimation," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1636–1646, 2018.

[3] S. L. Dhingra, P. P. Mujumdar, and Rajesh H. Gajjar, "Application of time series techniques for forecasting truck traffic attracted by the bombay metropolitan region," *Journal of Advanced Transportation*, vol. 27, no. 3, pp. 227–249, 1993.

[4] A. Dorestani Z. Rezaee and S. Aliabadi, "Application of time series analyses in big data: Practical, research, and education implications," *Allen Press*, vol. 15, November 2017.

[5] Pedro Lara-Benítez, Manuel Carranza-García, and José C. Riquelme, "An experimental review on deep learning architectures for time series forecasting," *International Journal of Neural Systems*, vol. 31, no. 03, pp. 2130001, 2021, PMID: 33588711.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. 2017, vol. 30, Curran Associates, Inc.

[7] Véronique Millette and Natalie Baddour, "Signal processing of heart signals for the quantification of non-deterministic events - biomedical engineering online," Jan 2011.

[8] Tai Sing Lee, "Image representation using 2d gabor wavelets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 959–971, 1996.

[9] Rob J. Hyndman and Anne B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.

[10] Bryan Lim, Sercan Ö. Arık, Nicolas Loeff, and Tomas Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021.

[11] Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio, "N-beats: Neural basis expansion analysis for interpretable time series forecasting," 2019.

[12] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski, "Deepar: Probabilistic forecasting with autoregressive recurrent networks," *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[13] Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C. Maddix, Syama Rangapuram, David Salinas, Jasper Schulz, Lorenzo Stella, Ali Caner Türkmen, and Yuyang Wang, "GluonTS: Probabilistic and Neural Time Series Modeling in Python," *Journal of Machine Learning Research*, vol. 21, no. 116, pp. 1–6, 2020.

[14] Skipper Seabold and Josef Perktold, "statsmodels: Econometric and statistical modeling with python," in *9th Python in Science Conference*, 2010.

[15] Google, "Web traffic time series forecasting," 2017.