Attention-Based Autoregression for Accurate and Efficient Multivariate Time Series Forecasting

Jaemin Yoo* U Kang*†

Abstract

Given a multivariate time series, how can we forecast all of its variables efficiently and accurately? The multivariate forecasting, which is to predict the future observations of a multivariate time series, is a fundamental problem closely related to many real-world applications. However, previous multivariate models suffer from large model sizes due to the inefficiency of capturing complex intra-variable patterns and inter-variable correlations, resulting in poor accuracy. In this work, we propose AttnAR (attention-based autoregression), a novel approach for general multivariate forecasting which maximizes its model efficiency via separable structure. AttnAR first extracts variable-wise patterns by a mixed convolution extractor that efficiently combines deep convolution layers and shallow dense layers. Then, AttnAR aggregates the patterns by learning time-invariant attention maps between the target variables. AttnAR accomplishes the state-of-the-art forecasting accuracy in four datasets with up to 117.3 times fewer parameters than the best competitors.

1 Introduction

Given a multivariate time series, how can we forecast all of its variables efficiently and accurately? The multivariate forecasting, which is to predict the future observations of a multivariate time series [9], is a fundamental problem that has been studied widely for various tasks [2, 16, 18]. For instance, when forecasting the electricity consumption of a city, it is essential to acquire the records of adjacent places for accurate predictions, since their consumptions are correlated due to the shared weather, culture, and infrastructure.

However, previous deep learning-based models [11, 16] for the problem suffer from large model sizes which lead to reasonable performance only if the hyperparameters are perfectly tuned for each dataset. This is because a single multivariate model aims to predict all target variables at the same time, where the number of variables varies with datasets from a few to hundreds. As a result, simple univariate models [5, 7, 14] are favored over multivariate models in many cases due to their robustness and consistency. On the other hand, traditional models [20] have small sizes, but are not as accurate as deep learning models since they focus only on the linear relationship between observations.

Specifically, the previous models have the following limitations that prevent them from achieving high accuracy of multivariate forecasting. First, they consist of complex mixtures of deep learning operations, making it difficult to tune optimal capacity for various datasets. Second, they focus on inter-variable relationships, losing complex variable-wise patterns that can be captured from the observation of each variable. Third, the correlations are captured only from the current observations, which constantly change over time having little consistency, decreasing the robustness of model.

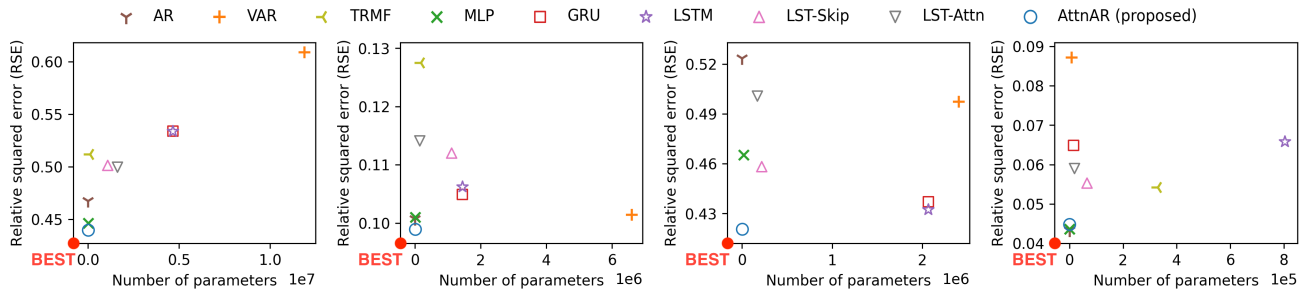
In this work, we propose AttnAR (attention-based autoregression), a novel approach for multivariate time series forecasting. AttnAR addresses the limitations of previous models by the following ideas. First, it consists of separable modules, each of which aims at a distinct objective: capturing variable-wise patterns, aggregating them, and making the predictions (Figure 2). Second, it captures complex patterns for each variable by combining deep convolution layers and shallow fully-connected layers (Figure 3). Third, it utilizes the intrinsic properties of data to learn time-invariant attention maps between the target variables (Figure 4).

Our contributions are summarized as follows:

- **Method.** We propose AttnAR, a novel approach for multivariate time series forecasting, which explicitly learns the correlations between variables as stable time-invariant attention maps.
- **Accuracy and parameter-efficiency.** AttnAR achieves the best accuracy with up to 117.3× fewer parameters than the previous state-of-the-art models (Figure 1) using the mixed convolution extractor and time-invariant attention which capture the intra- and inter-variable patterns, respectively.
- **Interpretability.** Time-invariant attention maps learned by AttnAR are directly interpretable, giving new insights on the given dataset.

*Computer Science and Engineering, Seoul National University (jaeminyoo@snu.ac.kr, ukang@snu.ac.kr).

†DeepTrade Inc.



(a) Traffic (b) Electricity (c) Solar-Energy (d) Exchange-Rate

Figure 1: The relative squared errors (RSE) of AttnAR and the baselines with respect to the number of parameters when the prediction horizon h is 24. Note that AttnAR gives the best performance, closest to the best point that represents the smallest error with the least number of parameters.

2 Related Works

We first define the problem of multivariate time series forecasting. Then, we introduce related works categorized as recurrent neural networks and matrix factorization, and describe the limitations of previous models.

2.1 Multivariate Forecasting We define the problem of multivariate forecasting as in [11]. Given a time series $\mathbf{X} \in \mathbb{R}^{d \times w}$ where d is the number of variables and w is the window size, the problem is to predict the values of all variables at a future time step $t+h$ as a vector $\hat{\mathbf{y}} \in \mathbb{R}^d$, where t is the prediction moment and h is called a horizon. The problem is more difficult with larger h, as the distance to the target time step increases.

Multivariate forecasting has been studied extensively for various real-world tasks [1, 15, 18]. The most typical approach is an autoregressive (AR) model which learns for each variable a linear mapping between the observed and target values. AR is considered as a strong baseline due to its robustness, especially on noisy data having no clear patterns. Vector autoregression (VAR) extends AR to consider multiple variables at the same time, but it easily overfits to training data due to many parameters. They are the simplest baselines on multivariate forecasting which rely on linear relationships.

2.2 Recurrent Neural Networks Recurrent neural networks (RNN) such as LSTM [8] and GRU [3] have been used widely for time series forecasting as representative deep learning models [5, 7, 14, 16]. However, RNN-based models require a large amount of parameters, especially when the number of target variables is large, because the state vector passed through RNN cells should be long enough to contain the information of all variables. If the length of state vectors is much smaller than the number d of variables, it is not possible to produce accurate predictions for all d variables considering the distinct property of each variable.

LSTNet [11] is a recent GRU-based approach that improves RNNs by skip connections and temporal attention between distant cells. It first decreases the length of input vectors by applying 2D convolutions to the data matrix before feeding them to the GRU cells. Then, it connects distant cells using skip connections or temporal attention, along with the local connections between adjacent cells. As a result, LSTNet achieved the state-of-the-art accuracy in real-world datasets.

However, LSTNet has introduced a large number of hyperparameters, and thus is vulnerable to a small change of input data or hyperparameters. For instance, one needs to manually choose the interval of skip connections by carefully analyzing the repetitive patterns of given time series. Other hyperparameters, such as the number of convolution channels, convolution width, and the length of GRU hidden states, also need to be adjusted carefully to produce its maximum performance.

Our AttnAR explicitly learns the correlations between target variables as an attention map, instead of passing long state vectors through the RNN cells that contain the information for all variables. This avoids the overfitting problem and takes the opportunity to incorporate multiple variables by parameter sharing, without a large number of hyperparameters as in LSTNet.

2.3 Matrix Factorization Models Matrix factorization (MF) is to approximate a matrix as the product of two low-rank matrices. Since a multivariate time series is represented as a matrix, MF has been applied to find low-dimensional embeddings of variables and time steps [19, 20]. The learned embeddings are directly used for multivariate forecasting, as they contain the essential properties of variables and time steps.

TRMF [20] is a recent MF-based approach that uses an AR model as a temporal regularizer to make adjacent

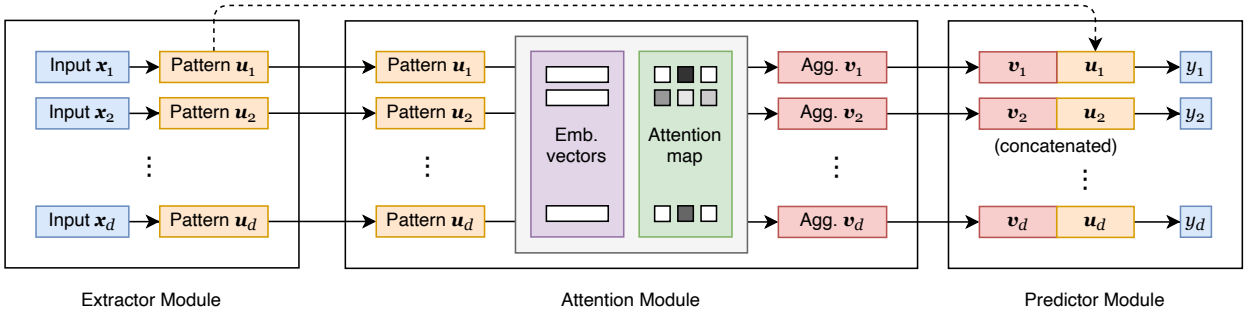


Figure 2: Overall architecture of AttnAR, consisting of *extractor*, *attention*, and *predictor* modules. AttnAR captures variable-wise patterns by the extractor module and then correlates them by the attention module, based on the learned embeddings of variables. Then, AttnAR produces the final output using the predictor module.

time steps have similar embeddings. In other words, TRMF learns the embeddings of time steps that the AR model is able to predict well, in addition to minimizing the reconstruction error between the original matrix and the product of learned embeddings. For inference at test time, TRMF first predicts the embedding of the target step by the trained AR model and multiplies it with the embeddings of variables to generate predictions.

However, MF-based models have two disadvantages that limit their performance on time series forecasting. First, they assume a simple linear relationship between observations and embedding vectors, which is not true in many real-world datasets. Second, as they learn embedding vectors for all time steps, a model requires keeping the embedding vectors at all prediction steps even though the observation is given. This also makes it difficult to find optimal hyperparameters, since a trained model works only once with its maximum performance; it should be trained n times for n validation steps.

We get the idea of embeddings from the MF-based approaches, since it is an efficient way of learning the intrinsic properties of variables as numerical vectors. Still, we address the limitations of MF-based approaches by a) using the embedding vectors in a non-linear way and b) learning only the embeddings of target variables, ignoring time steps, to avoid inefficient retraining. Thus, our method produces robust and consistent predictions based on static embeddings once it is trained.

3 Proposed Approach

We propose AttnAR (attention-based autoregression), a novel approach that learns the correlations between variables for accurate time series forecasting.

3.1 Overview The following challenges need to be addressed for accurate multivariate forecasting:

- **Flexible model capacity.** How can we adjust the complexity of a prediction model for each dataset without requiring many hyperparameters?
- **Complex input patterns.** How can we capture complex variable-wise patterns separately from the correlations between variables?
- **Intrinsic properties.** How can we capture the intrinsic properties of variables for learning accurate correlations between variables?

AttnAR addresses the aforementioned challenges by the following main ideas:

- **Separable modules.** We design AttnAR to have separable modules each of which aims at a specific objective, making it easy to tune its capacity based on the property of each dataset (Figure 2).
- **Mixed convolution extractor.** We use a mixed convolution extractor to capture complex variable-wise patterns, which consists of deep convolutions and shallow fully-connected layers (Figure 3).
- **Time-invariant attention.** We learn the intrinsic properties of variables as embedding vectors and make time-invariant attention maps which produce consistent and interpretable results (Figure 4).

AttnAR consists of three components as illustrated in Figure 2. The extractor first captures a variable-wise pattern from the observation of each variable. Then, the attention module correlates the extracted patterns by using an attention map of the target variables, which is calculated based on the intrinsic properties of variables. The predictor takes the concatenation of the variable-wise and aggregated patterns as an input to produce the final predictions for all variables.

3.2 Mixed Convolution Extractor The extractor of AttnAR captures nonlinear patterns from the observation of each variable. Given an input vector $\mathbf{x}_i$ which contains w observations of variable i, we apply an ex-

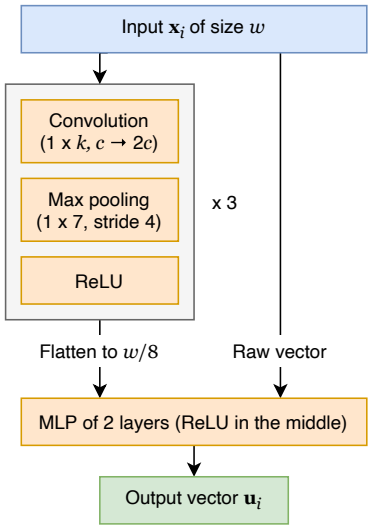


Figure 3: The structure of our mixed convolution extractor, which captures short-term complex patterns by deep convolution layers and long-term straightforward patterns by shallow fully-connected layers.

tractor function f to produce a pattern vector $\mathbf{u}_i$:

$$\mathbf{u}_i = f(\mathbf{x}_i; \theta_f), \tag{3.1}$$

where θ_f is the set of learnable parameters in f.

The simplest choice of f for capturing a nonlinear pattern is a multilayer perceptron (MLP) with a single hidden layer, which is defined as follows:

$$f_{\text{mlp}}(\mathbf{x}_i) = \mathbf{W}^{(2)}\text{ReLU}(\mathbf{W}^{(1)}\mathbf{x}_i + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)}, \tag{3.2}$$

where $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are learnable parameters of the l-th layer, and the ReLU activation [12] is used.

However, the MLP extractor of Equation (3.2) lacks the ability of capturing complex patterns which require deep layers to be detected. It is possible to increase the number of layers of the MLP extractor, but it increases the number of parameters redundantly. Thus, we stack deep layers with only a few additional parameters by our *mixed convolution extractor* f_{mce}, utilizing 1D convolutions to leverage the temporal locality of observations:

$$f_{\text{mce}}(\mathbf{x}_i) = f_{\text{mlp}}(\mathbf{x}_i \parallel f^n_{\text{conv}}(\mathbf{x}_i)), \tag{3.3}$$

where f^n_{conv} represents a series of n convolution blocks, $\parallel$ represents the concatenation of two vectors, and f_{mlp} represents the MLP module of Equation (3.2).

The structure of our extractor is depicted in Figure 3. It contains three convolution blocks each of which contains a $1 \times k$ convolution layer, a 1×7 max pooling layer with the stride of 4, and the ReLU activation.[1] At each convolution, we add zero padding at both ends of input to preserve its length and increase the number of channels twice. The result after the three convolution blocks is flattened and then concatenated with the raw input $\mathbf{x}_i$ before being fed into the MLP.

Stacking deep dense layers requires a large number of parameters, leading to overfitting in time series forecasting. Our convolution operations make it possible to capture complex nonlinear patterns using a negligible number of additional parameters. For instance, the convolution blocks require less than 300 parameters when $k = 7$, while a single fully-connected layer requires $4,096 = 64^2$ parameters when each layer has 64 units.

3.3 Time-Invariant Attention Our objective is to combine the extracted patterns of multiple variables to capture rich information that cannot be inferred from the observation of a single variable. We first introduce three approaches for learning an attention map between variables, which differ from each other by how to choose a query, keys, and values. Then, we show that our time-invariant attention is the most effective for multivariate forecasting due to its robustness and consistency.

We summarize the three approaches as follows:

1. **Basic attention:** We consider a pattern vector $\mathbf{u}_i$ as a query, a key, and a value.
2. **Hybrid attention:** We learn an embedding vector $\mathbf{h}_i$ of variable i, which does not depend on $\mathbf{u}_i$, and use it as a query. We use $\mathbf{u}_i$ as a key and a value.
3. **Time-invariant attention:** We learn $\mathbf{h}_i$ as in the hybrid approach but consider it both as a query and a key, while using $\mathbf{u}_i$ as a value.

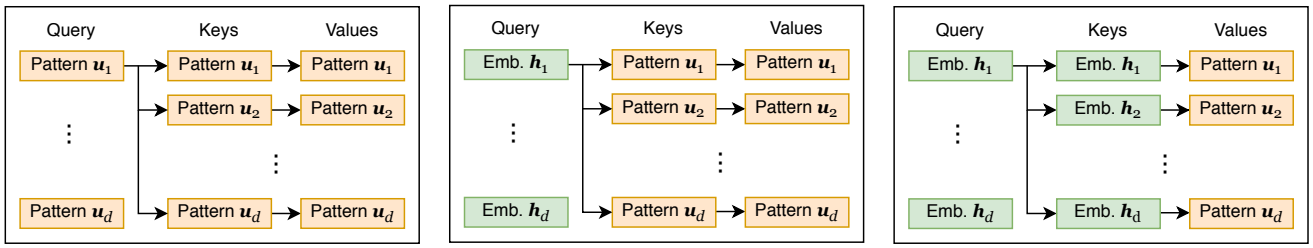
Figure 4 compares the three approaches. They have different degrees of freedom of adjusting attention maps to the pattern vectors. The basic attention determines its attention map solely from the pattern vectors, while the hybrid and time-invariant attentions use the variable embeddings as queries.

We briefly introduce the attention [17], which is a general technique that takes a query, keys, and values as input to compute a weighted average of the values. Consider n pairs of key and value vectors $\{(\mathbf{k}_i, \mathbf{e}_i) \mid i = 1, ..., n\}$. Given a query vector $\mathbf{q}$ of the same length as the keys, the result of attention is given as follows:

$$\text{attention}(\mathbf{q}) = \frac{\sum_i \exp(\mathbf{q}^\top \mathbf{k}_i)\mathbf{e}_i}{\sum_j \exp(\mathbf{q}^\top \mathbf{k}_j)}. \tag{3.4}$$

In other words, the attention gives larger weights to the value vectors whose key vectors produce larger dot products with the query, where all weights always sum to one. The attention mechanism has been adopted in various models [4, 6] due to its simplicity.

[1] k is chosen between 3, 5, and 7 in our experiments.



(a) Basic attention (Section 3.3.1). (b) Hybrid attention (Section 3.3.2). (c) Time-invariant attention (S. 3.3.3).

Figure 4: Illustrations of our attention approaches: (a) *basic*, (b) *hybrid*, and (c) *time-invariant* attentions. They differ from each other by how to use the pattern vectors $\mathbf{u}_i$ and embedding vectors $\mathbf{h}_i$. The attention map of the time-invariant approach remains the same after the training, while those of the others change over time.

3.3.1 Basic Attention The simplest approach is to use $\mathbf{u}_i$ as the query, key, and value of each variable i. This finds target variables whose pattern vectors are similar to $\mathbf{u}_i$ and then combines their observations to improve the accuracy of prediction for variable i.

The main limitation of basic attention is that the correlations depend only on the current patterns, ignoring the intrinsic properties of data. It is likely that the output remains similar even with the attention, since it aggregates the pattern vectors that are already similar to $\mathbf{u}_i$. As a result, it cannot find variables which *should* produce similar outputs even though their current patterns are different; for instance, if two cities show different patterns temporarily although they are in the same area, it is desirable to correlate them.

3.3.2 Hybrid Attention Our second approach is to learn the embedding vectors of variables and use them as the queries of attention. We initialize randomly an embedding vector $\mathbf{h}_i$ for each variable i as a trainable parameter and update it through backpropagation. This enables us to learn the static property of each variable as in MF-based models [19, 20], separately from the pattern vectors that change constantly by input.

The hybrid approach computes the aggregated vector $\mathbf{v}_i$ for variable i as follows:

$$\mathbf{v}_i = \frac{\sum_k \exp(\mathbf{h}_i^\top \mathbf{u}_k)\mathbf{u}_k}{\sum_j \exp(\mathbf{h}_i^\top \mathbf{u}_j)}. \tag{3.5}$$

which gives a large weight to $\mathbf{u}_k$ that produces a large dot product with $\mathbf{h}_i$. The static and dynamic properties of variable i are captured by $\mathbf{h}_i$ and $\mathbf{u}_i$, respectively.

The main limitation of the hybrid attention is that the pattern vectors become dependent on the attention. Each pattern vector $\mathbf{u}_i$ is learned to be meaningful with regard to the attention and prediction at the same time, increasing the difficulty of training. On the other hand, the embedding $\mathbf{h}_i$ should be similar to pattern vectors to find proper attention maps, which is problematic if input observations change dramatically over time.

3.3.3 Time-Invariant Attention Our main attention function learns a static attention map that is not affected by the pattern vectors. We learn an embedding vector $\mathbf{h}_i$ for each variable i as in the hybrid approach, but use it both as a query and a key. That is, the attention map is determined solely by the intrinsic properties of variables, making a model robust and consistent regardless of the change of input patterns. This is advantageous in multivariate forecasting, where it is important to learn *inherent* relationships between variables.

One notable advantage of the time-invariant attention is that an interpretable attention map is given after the training, which gives valuable information about the target variables on how much they are correlated to each other. Furthermore, since it learns asymmetric relationships, the attention map tells us which variables precede the others in terms of multivariate forecasting.

The performance of time-invariant attention can be easily improved if additional information is given. The random initialization of $\mathbf{h}_i$ represents that we have no prior knowledge of the dataset. If we have an attribute vector for each variable, we can design a parameterized function that maps such attributes to $\mathbf{h}_i$. Nevertheless, we use the simple embedding in this work, as we focus on time series having no additional input features.

3.4 Predictor Module The predictor module takes the concatenation of the variable-wise pattern $\mathbf{u}_i$ and the aggregated pattern $\mathbf{v}_i$ of variable i to produce the prediction $\hat{y}_i$. We use a simple MLP of Equation (3.2), but the output is a scalar rather than a vector:

$$\hat{y}_i = f_{\text{mlp}}(\mathbf{u}_i \parallel \mathbf{v}_i). \tag{3.6}$$

We concatenate $\mathbf{v}_i$ to $\mathbf{u}_i$, since $\mathbf{v}_i$ contains little information of variable i after the attention; we lose the

Table 1: Summary of datasets.[2]

Dataset	Length	Dim.	Granularity
Traffic	17,544	862	1 hour
Electricity	26,304	321	1 hour
Solar-Energy	52,560	137	10 minutes
Exchange-Rate	7,587	8	1 day

core information of variable i if using only $\mathbf{v}_i$ as the input of the predictor module. We use a simple predictor intentionally, since the previous modules impose sufficient nonlinearity to the pattern vectors. Nevertheless, we use an MLP instead of linear autoregression, because the complex relationship between variable-wise and aggregated patterns can be captured only by a nonlinear function with sufficient capacity.

4 Experimental Settings

We present datasets, competitors, and hyperparameters for our experiments. All of our experiments were done at a workstation with GTX 1080 Ti.

4.1 Datasets We use four time series datasets that have been used in [11], whose information is summarized in Table 1. Traffic is an hourly data from the California Department of Transportation, which describes the road occupancy rates as numerical values between 0 and 1. Electricity is electricity consumption recorded at every hour for 321 clients. Solar-Energy contains solar power production records, sampled at every 10 minutes from 137 plants. Exchange-Rate is a collection of the daily exchange rates of eight countries from 1990 to 2016. The preprocessed datasets are publicly available.[2]

4.2 Competitors We compare our approach with the following baselines. An autoregressive (AR) model is the simplest approach that learns a linear function between the previous and future observations. Vector autoregression (VAR) extends AR using all variables for multivariate forecasting. Temporal regularized matrix factorization (TRMF) is the state-of-the-art MF-based approach for time series forecasting. We also consider a multilayer perceptron (MLP) as a nonlinear baseline.

We consider gated recurrent units (GRU) and long short-term memory units (LSTM) as baselines of deep learning-based models. LSTNet [11] is the state-of-the-art model for multivariate time series forecasting, which combines convolutional and recurrent neural networks for long- and short-term correlations. We use LST-Attn and LST-Skip, the two different versions of LSTNet that adopt different approaches to capture long-term correlations that the RNN cannot detect, as competitors.

[2]https://github.com/laiguokun/multivariate-time-series-data

Table 2: Comparison of various extractor and attention modules of AttnAR, where h denotes the horizon. The mixed convolution extractor (MCE) and time-invariant attention (TIA) show the best performance.

Extractor+Attn. of AttnAR	Traffic (RSE)		Solar (RSE)	
	h=6	h=24	h=6	h=24
MLP (baseline)	.4368	.4464	.2747	.4652
MLP + Basic	.4387	.4492	.2422	.4730
MLP + Hybrid	.4304	.4466	.2461	.4379
MLP + TIA	**.4287**	.4442	.2332	.4326
MCE + TIA	**.4287**	**.4396**	**.2272**	**.4205**

Extractor+Attn. of AttnAR	Traffic (COR)		Solar (COR)	
	h=6	h=24	h=6	h=24
MLP (baseline)	.8824	.8767	.9618	.8813
MLP + Dynamic	.8799	.8758	.9685	.8762
MLP + Hybrid	.8848	.8762	.9694	.8966
MLP + TIA	.8859	.8775	.9727	.8982
MCE + TIA	**.8865**	**.8800**	**.9741**	**.9031**

4.3 Hyperparameters We split each dataset into training, validation, and testing by the 6:2:2 ratio with the chronological order. We z-normalize each dataset by calculating the average and standard deviation over training data, and applying them for all data. This is to make sure that only training data have been observed at the time of experiments. We have implemented our AttnAR and all baselines using PyTorch [13]. We use the Adam optimizer [10] to minimize the mean squared error (MSE) loss and stop the training if the validation loss does not decrease for 10 epochs.

We perform a grid search to find optimal hyperparameters of all methods that minimize the validation errors. The window size w is searched in $\{2^0, 2^1, ..., 2^9\}$ for each model and dataset. For GRU and LSTM, we search the numbers of units and layers in $\{2^5, ..., 2^9\}$ and $\{1, 2\}$, respectively. For LSTNet, we search its hyperparameters as in the original paper [11]. For AttnAR, we search the embedding size and the number of hidden units of MLP independently in $\{2^3, 2^4, 2^5\}$. We use the mixed convolution extractor and time-invariant attention as our default choices unless stated otherwise.

4.4 Evaluation Metrics We adopt two evaluation metrics as in [11]. Consider an answer matrix $\mathbf{Y}$ whose size is $d \times n$, where d is the number of variables and n is the length of test data. We also assume a prediction matrix $\hat{\mathbf{Y}}$ having the same size, which is generated by a

Table 3: RSEs (the lower the better) of AttnAR and the baselines, where h represents the horizon of predictions. AttnAR achieves the lowest errors in all cases except in Exchange-Rate, where AR and MLP work the best.

Method	Traffic			Electricity			Solar-Energy			Exchange-Rate		
	h=6	h=12	h=24	h=6	h=12	h=24	h=6	h=12	h=24	h=6	h=12	h=24
AR	.4647	.4659	.4675	.0930	.0983	.1007	.3120	.4195	.5235	**.0238**	.0329	**.0433**
VAR	.5909	.6008	.6088	.0964	.1010	.1014	.2965	.4112	.4974	.0496	.0652	.0872
TRMF	.4871	.4909	.5120	.1050	.1062	.1275	.6001	.7112	.8434	.0425	.0466	.0542
MLP	.4368	.4436	.4464	**.0871**	.0965	.1010	.2747	.3592	.4652	**.0238**	**.0328**	.0436
GRU	.5158	.5225	.5340	.1088	.0974	.1049	.2485	.3229	.4370	.0322	.0465	.0639
LSTM	.5195	.5268	.5337	.1043	.1008	.1062	.2539	.3328	.4323	.0412	.0503	.0658
LST-Skip	.4811	.4900	.5013	.0993	.0959	.1120	.2537	.3448	.4582	.0279	.0425	.0553
LST-Attn	.4780	.4895	.4996	.0936	.0990	.1141	.2552	.3528	.5007	.0379	.0473	.0590
AttnAR	**.4287**	**.4370**	**.4396**	**.0871**	**.0942**	**.0989**	**.2272**	**.3057**	**.4205**	.0240	.0336	.0448

model by stacking n predictions. Then, the root relative squared error (RSE) is defined as follows:

$$\text{RSE}(\hat{\mathbf{Y}}, \mathbf{Y}) = \frac{||\mathbf{Y} - \hat{\mathbf{Y}}||_{\text{F}}}{||\mathbf{Y} - \text{avg}(\mathbf{Y})||_{\text{F}}}, \tag{4.7}$$

where $||\cdot||_{\text{F}}$ represents the Frobenius norm.

Another evaluation metric that we use is an empirical coefficient (COR) that is defined as follows:

$$\frac{1}{d}\sum_{i=1}^{d} \frac{(\mathbf{y}_i - \text{avg}(\mathbf{y}_i)\mathbf{1})^\top(\hat{\mathbf{y}}_i - \text{avg}(\hat{\mathbf{y}}_i)\mathbf{1})}{\text{std}(\mathbf{y}_i)\text{std}(\hat{\mathbf{y}}_i)}, \tag{4.8}$$

where $\text{std}(\mathbf{x}) = \sqrt{\sum_i (x_i - \text{mean}(\mathbf{x}))^2}$, and $\mathbf{y}_i$ and $\hat{\mathbf{y}}_i$ are the true and the prediction vectors that correspond to variable i, respectively. We multiply $\text{avg}(\mathbf{y}_i)$ with the one vector $\mathbf{1}$ to subtract it from every element of $\mathbf{y}_i$.

COR values evaluate how much the predictions move along with the observations, rather than directly comparing the raw values. Simple models are expected to give higher COR values compared to complex models, since they are robust to noisy inputs; complex models are easily affected by small changes of input and produce inconsistent predictions. Thus, we evaluate the prediction ability and robustness of a model at the same time by using these two evaluation metrics.

5 Experimental Results

We aim to answer the following questions:

- **Q1 Accuracy (Sec. 5.1).** How accurate is AttnAR compared to the previous approaches for multivariate forecasting? Does it make consistent improvements with both RSE and COR?
- **Q2 Ablation study (Sec. 5.2).** Which extractor and attention modules should we choose? Do our mixed convolution extractor and time-invariant attention help improve the accuracy?
- **Q3 Parameter-efficiency (Sec. 5.3).** Can AttnAR be more parameter-efficient than the previous models, while achieving a higher accuracy?
- **Q4 Interpretability (Sec. 5.4).** How can we interpret the attention maps learned by AttnAR to understand the unique properties of datasets?

5.1 Accuracy We compare the prediction accuracy of AttnAR and baselines in Tables 3 and 4 by the RSE and COR, respectively. AttnAR is more accurate than all the baselines except in the Exchange-Rate dataset which is too noisy and thus AR and MLP work generally the best by focusing on each variable independently. Even in this dataset, AttnAR shows competitive results that outperform all the other baselines.

In Electricity, the RNN-based multivariate models, including both LST-Skip and LST-Attn, produce poor CORs compared to the RSE. This demonstrates the difficulty of making consistent predictions with correlating the target variables, which AttnAR effectively addresses by the three main ideas: separable modules, mixed convolution extractor, and time-invariant attention.

5.2 Ablation Study Table 2 compares various attention and extractor modules of AttnAR in the Traffic and Solar-Energy datasets. We include the MLP as a baseline that does not correlate the variables. AttnAR with the mixed convolution extractor (MCE) and the time-invariant attention (TIA) performs the best among all five settings with consistent improvements.

Comparing the three attention modules, we observe that the accuracy improves by adding static properties to the attention: TIA > hybrid > basic. This shows the importance of robust correlations for multivariate forecasting. The basic attention works worse than the MLP in some cases, as it can correlate wrong variables if given noisy observations. At the same time, our MCE

Table 4: CORs (the higher the better) of AttnAR and the baselines, where h represents the horizon of predictions. AttnAR produces the largest correlations in most cases, demonstrating the robustness of its predictions.

Method	Traffic			Electricity			Solar-Energy			Exchange-Rate		
	h=6	h=12	h=24	h=6	h=12	h=24	h=6	h=12	h=24	h=6	h=12	h=24
AR	.8674	.8660	.8646	.9050	.8903	.8877	.9492	.9039	.8429	**.9673**	.9520	.9325
VAR	.7798	.7641	.7754	.6691	.6274	.8290	.9566	.9096	.8575	.7263	.7033	.6794
TRMF	.8529	.8504	.8384	.8138	.8092	.8006	.8100	.7108	.5194	.8535	.8419	.8196
MLP	.8824	.8785	.8767	**.9170**	.8812	.8847	.9618	.9329	.8813	**.9673**	.9520	**.9390**
GRU	.8540	.8493	.8412	.6098	.6170	.7438	.9702	.9470	.8996	.7980	.6731	.6923
LSTM	.8515	.8491	.8475	.6952	.7079	.7494	.9684	.9442	.9012	.6404	.6276	.5721
LST-Skip	.8670	.8627	.8570	.6365	.7270	.5609	.9676	.9384	.8862	.8125	.7915	.7662
LST-Attn	.8681	.8634	.8579	.7535	.7812	.8105	.9670	.9374	.8631	.8973	.8547	.7972
AttnAR	**.8865**	**.8819**	**.8800**	.9160	**.9108**	**.9089**	**.9741**	**.9519**	**.9031**	.9672	**.9536**	.9248

Table 5: Numbers of parameters of AttnAR and RNN-based models when the horizon $h = 24$. AttnAR has up to 42.6× fewer parameters than the competitors.

Method	Traffic	Elec.	Solar	Exchange
GRU	4665.3K	1445.4K	2066.9K	14.5K
LSTM	4665.3K	1445.4K	2066.9K	804.4K
LST-Skip	1086.1K	1114.1K	218.7K	65.4K
LST-Attn	1621.5K	144.1K	170.5K	18.6K
AttnAR	**25.5**K	**9.5**K	**10.7**K	**0.9**K

outperforms the MLP extractor by effectively capturing complex patterns using deep convolutions.

5.3 Parameter-Efficiency Figure 1 compares RSEs of various approaches with respect to the number of parameters. AttnAR achieves the state-of-the-art performance with few parameters, as it is closest to the best point in all datasets. A notable observation is that the number of parameters and the error are correlated positively in some cases, especially in the Traffic dataset, representing that using more parameters decreases the performance. This is due to the property of multivariate forecasting, where a model overfits easily.

Table 5 reports the exact number of parameters of each model in the same experiment, comparing AttnAR with the RNN-based methods. AttnAR contains fewer parameters than all competitors, which is up to 117.3× and 63.6× fewer than those of LST-Skip and LST-Attn, respectively, which are the state-of-the-art models. The table also shows the sensitivity of LSTNet to the hyperparameters. LST-Skip contains 7.3× more parameters than LST-Attn at Electricity, because of the difference of hyperparameters tuned for validation data; e.g., the number of parameters of each model is largely affected by the length of RNN state vectors.

5.4 Interpretability Attention maps learned from AttnAR are directly interpretable and produce valuable information about the properties of datasets. Figure 5 shows the attention maps for all datasets when $h = 24$. Solar-Energy shows the largest correlations between the variables, Traffic and Electricity show weak correlations, and Exchange-Rate shows no correlations.

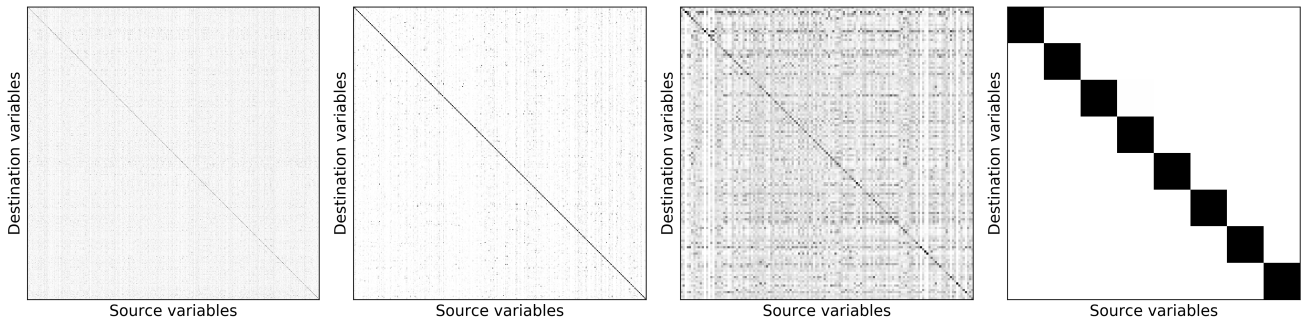
The distinct property of Solar-Energy is explained by the learned attention map demonstrating high correlations between variables. In the experiments on Solar-Energy of Table 3, RNN-based methods outperform AR and MLP by large margins, which predict each variable independently. In Table 2, the difference between attention methods is larger in Solar-Energy than in Traffic. This is because the variables in Solar-Energy are highly correlated to each other, and it is important to capture the correlations for achieving high accuracy.

6 Conclusion

We propose AttnAR (attention-based autoregression), a novel approach for multivariate time series forecasting which maximizes its parameter-efficiency by a separable structure. AttnAR first extracts variable-wise patterns by a mixed convolution extractor that consists of deep convolution layers and shallow dense layers. AttnAR then aggregates the captured patterns by time-invariant attention which produces stable and consistent results. As a result, AttnAR achieves the state-of-the-art accuracy while requiring up to 117.3× fewer parameters than the best competitors. Moreover, the learned attention maps between variables are directly interpretable, giving us rich information to understand the properties of given time series data even with no prior knowledge.

Acknowledgments

This work is supported by DeepTrade Inc.



(a) Traffic (b) Electricity (c) Solar-Energy (d) Exchange-Rate

Figure 5: Attention maps learned from AttnAR with the time-invariant attention. We map each element x to $x^{1/3}$ before applying linear color maps to clearly visualize the correlations. The element (u, v) of each map represents the amount of influence from variable v to variable u. The correlations between variables are the most distinct in (c) Solar-Energy, weak in (a) Traffic and (b) Electricity, and non-existing in (d) Exchange-Rate.

References

[1] B. O. Barnett, *An introduction to time series forecasting for CPE*, in 17th International Computer Measurement Group Conference, 1991, pp. 1197–1206.

[2] T. Chen, H. Yin, H. Chen, L. Wu, H. Wang, X. Zhou, and X. Li, *TADA: trend alignment with dual-attention multi-task recurrent neural networks for sales prediction*, in ICDM, 2018.

[3] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, *Learning phrase representations using RNN encoder-decoder for statistical machine translation*, in EMNLP, 2014.

[4] J. Devlin, M. Chang, K. Lee, and K. Toutanova, *BERT: pre-training of deep bidirectional transformers for language understanding*, in NAACL-HLT, 2019.

[5] V. Flunkert, D. Salinas, and J. Gasthaus, *Deepar: Probabilistic forecasting with autoregressive recurrent networks*, CoRR, abs/1704.04110 (2017).

[6] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, *Dual attention network for scene segmentation*, in CVPR, 2019, pp. 3146–3154.

[7] J. Gasthaus, K. Benidis, Y. Wang, S. S. Rangapuram, D. Salinas, V. Flunkert, and T. Januschowski, *Probabilistic forecasting with spline quantile function rnns*, in AISTATS, 2019.

[8] S. Hochreiter and J. Schmidhuber, *Long short-term memory*, Neural Computation, 9 (1997).

[9] J. Jang, D. Choi, J. Jung, and U. Kang, *Zoom-svd: Fast and memory efficient method for extracting key patterns in an arbitrary time range*, in CIKM, 2018.

[10] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, in ICLR, 2015.

[11] G. Lai, W. Chang, Y. Yang, and H. Liu, *Modeling long- and short-term temporal patterns with deep neural networks*, in SIGIR, 2018.

[12] V. Nair and G. E. Hinton, *Rectified linear units improve restricted boltzmann machines*, in ICML, J. Fürnkranz and T. Joachims, eds., Omnipress, 2010.

[13] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, *Automatic differentiation in PyTorch*, in NIPS Autodiff Workshop, 2017.

[14] S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski, *Deep state space models for time series forecasting*, in NeurIPS, 2018.

[15] A. Sfetsos and A. Coonick, *Univariate and multivariate forecasting of hourly solar radiation with artificial intelligence techniques*, Solar Energy, 68 (2000).

[16] S. Shih, F. Sun, and H. Lee, *Temporal pattern attention for multivariate time series forecasting*, Mach. Learn., 108 (2019), pp. 1421–1441.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *Attention is all you need*, in NIPS, 2017.

[18] C. Voyant, M. Muselli, C. Paoli, and M.-L. Nivet, *Optimization of an artificial neural network dedicated to the multivariate forecasting of daily global radiation*, Energy, 36 (2011), pp. 348–359.

[19] L. Xiong, X. Chen, T. Huang, J. G. Schneider, and J. G. Carbonell, *Temporal collaborative filtering with bayesian probabilistic tensor factorization*, in SDM, 2010.

[20] H. Yu, N. Rao, and I. S. Dhillon, *Temporal regularized matrix factorization for high-dimensional time series prediction*, in NIPS, 2016.