





Knowledge Forcing: Fusing Knowledge-Driven Approaches with LSTM for Time Series Forecasting

Muhammad Ali Chattha^{1,2,3}(✉) , Muhammad Imran Malik³,
Andreas Dengel^{1,2} , and Sheraz Ahmed²

¹ Rheinland-Pfälzische Technical University Kaiserslautern-Landau, RPTU,
Kaiserslautern 67663, Germany
muhammad.ali.chattha@dfki.de

² German Research Center for Artificial Intelligence, DFKI,
Kaiserslautern 67663, Germany

³ National University of Science and Technology, NUST, Islamabad 44000, Pakistan

Abstract. Long Short-Term Memory (LSTM) typically relies solely on historical data for training and although, they excel at modelling sequential series and finding hidden patterns in the data, they are unable to utilize expert knowledge. Knowledge-driven systems (KDS), on the other hand, rely on domain knowledge and consist of rules explicitly defined by human experts. Both LSTM and KDS offer unique advantages, hence relying on a single approach can be suboptimal. However, currently there is a lacking of frameworks that can concurrently utilize explicit information in the KDS and hidden features in the data. In this paper, we propose a novel fusion mechanism, knowledge-forced LSTM (KF-LSTM), that combines knowledge-driven approaches with LSTM for time series forecasting. KF-LSTM employs LSTM in an encoder-decoder setting, where the decoder utilizes KDS predictions in a residual connection. This enables the decoder to utilize sequential relations in the historical data passed on by the encoder as well as information present in KDS in a complementary manner. We tested KF-LSTM on 4 real-world datasets in a multi-horizon forecasting setting. Even with utilizing relatively shallow single layered LSTM, KF-LSTM achieves State-of-the-Art (SotA) performance on almost all of the datasets, highlighting the information fusion capabilities of the framework. On average, knowledge forcing improves over previous SotA by **20%**.

Keywords: Time series forecasting · Knowledge fusion · Knowledge-driven system · Neural Networks · Hybrid Systems

1 Introduction

Time series forecasting is an important problem as it has great impact on many crucial domains such as demand prediction, financial forecasts, traffic flow prediction, weather forecasts, etc. Having an accurate estimate of future prospects

allows for better planning, which is pivotal for efficient resource management and profit generation. As a result, a great deal of importance is laid on time series forecasting and any improvements in forecasting approaches are highly sort after.

Time series forecasting approaches can be broadly categorized into two main categories: Knowledge-driven and data-driven approaches. Knowledge-Driven Systems (KDS) consists of explicitly defined rules that are made up by human experts who have substantial domain knowledge about the problem. These rules make up the knowledge base of the KDS and are used in a predefined manner for inference. As a result, KDS system can utilize the knowledge of human experts when performing the forecasting task. These expert rules are typically in the form of logical expressions such as first order logic [18,32], or mathematical expressions such as statistical methods [2,9]. Statistical methods normally consist of mathematical operations predefined by human experts and have shown great performance [16,17]. In contrast, data-driven methods rely on historical data from which they learn to extract hidden patterns that are useful for the solution. Long Short-Term Memory (LSTM) have shown considerable efficacy in modelling time series and sequential data. Both of the approaches, although used for the same goal, operate on very different underlying information. Both KDS and LSTM have different and unique strengths, and relying on a single approach can be suboptimal.

However, there is a severe lacking of frameworks that can combine the strengths of both KDS and LSTM. Current hybrid methods, for forecasting problem, mostly rely on ensemble methods, where two or more methods are combined after the inference by taking the average of their individual predictions. While ensemble methods do improve the overall result in some scenarios, they are limited by the accuracy of individual models [26]. Any inaccurate model can negatively impact the performance of the overall ensemble since, regardless of their accuracy, every model contributes to the final prediction. We believe that in an ideal fusion mechanism, the framework should be aware of the strengths of constituent models and should utilize these strengths accordingly, rather than simply taking an average.

Based on the above motivation, we propose a novel fusion framework, knowledge-forced LSTM (KF-LSTM), that fuses knowledge-driven approaches with LSTM in a way where their strengths are combined, and individual inaccuracies are catered for. KF-LSTM framework accesses the efficacy of predictions given by KDS and the utilizes information present in the historical data to offset any missing or incorrect information. This is achieved by using an encoder-decoder LSTM architecture, where KDS predictions are connected in a residual connection setting with the decoder. As a result, the decoder takes in information from KDS via the skip connection and information from historical data via the internal states of the encoder. Since the decoder calculates the residual function, it can offset and correct any missing or inaccurate information in the KDS, which current ensemble based hybrid methods are incapable of. We test knowledge forced LSTM on 4 real world forecasting datasets in a multi-horizon

forecasting setting. Although, we utilized a single layered LSTM model, the proposed framework outperformed recent transformer based forecasting models and achieved SotA performance across most of the datasets. On average, knowledge forcing framework improved over previous SotA by **20%**. In particular contribution of this paper are as follows:

- We introduce a novel fusion framework, KF-LSTM, that combines knowledge-driven approaches with LSTMs in a constructive manner.
- KF-LSTM achieves **20%** relative performance improvement over previous SotA on 4 real-world benchmark datasets.
- We show that KF-LSTM is agnostic to underlying KDS and can work with wide array KDS approaches.
- We show that KF-LSTM dynamically combines information from constituent models based on their individual efficacy.
- We show that KF-LSTM outperforms current ensemble based hybrid frameworks, highlighting its superior information fusion capabilities.

2 Related Work

In the context of time series forecasting, Hybrid schemes mostly revolve around ensemble-based methods. Syml et al. [22] utilized ensemble of DNN models with different parameters to obtain final forecast. Similarly, Larrea et.al. and Kaushik et. al. [10,13] also employed ensemble technique for the forecasting task. Ensemble technique combines different models in time series forecasting, but issues such as model diversity and accuracy can affect overall predictions. Highly inaccurate models can negatively effect the accuracy of the overall ensemble. Recently, attention-based models such as transformer networks, have gained interest for better forecasting performance [14,27,28]. The attention mask is computed by utilizing covariates in the dataset, which can be considered as knowledge from an additional source. Graph-based forecasting network aims to capture spatial dependencies among different time series in the dataset along with temporal modelling [6]. Both attention maps and spatial information can be considered as additional information that improve the overall forecasts; however, such techniques suffer when dealing with data that lacks mutual dependence or spatial information among time series.

Incorporating logic rules directly into neural network architecture has also been proposed [24,25]. Here, elements of the rule-set are considered as a unit in the neural network and their weights are pre-computed using relations defined in the rule-set. Some additional units that are not part of the logic rule-set are also used in the neural network to learn relations from the data as well. Such methods incorporate the knowledge base directly into the model, but this also limits flexibility and requires strong hierarchical coherence between the rule base and neural network layers. Although knowledge distillation is not directly applicable to the forecasting problem, it is still worth mentioning as a knowledge sharing framework [7,29]. Knowledge distillation is used for knowledge transfer, specifically in classification tasks. The method involves training a smaller model

(student network) to mimic the predictions of a larger, more complex model (teacher network). Although it improves the classification capabilities of student model, however, scenarios when teacher network is inaccurate is not catered.

3 Multi Horizon Forecasting

The main objective of a forecasting framework is to learn a parametric function that maps \mathbf{X}_w values from the past to $\hat{\mathbf{Y}}_h$, where w and h represents input window size and the output size, horizon, respectively. \mathbf{X}_w represents list of values $x_t, x_{t-1}, \dots, x_{t-w}$ and $\hat{\mathbf{Y}}_h$ represents list containing h predicted values $\hat{x}_{t+1}, \hat{x}_{t+2}, \dots, \hat{x}_{t+h}$. This can be mathematically expressed as:

$$[\hat{x}_{t+1}, \hat{x}_{t+2}, \dots, \hat{x}_{t+h}] = \Phi([x_t, x_{t-1}, \dots, x_{t-w}]; \mathcal{W}) \quad (1)$$

$$\hat{\mathbf{Y}}_h = \Phi(\mathbf{X}_w; \mathcal{W}) \quad (2)$$

where $\mathcal{W} = \{W_l, b_l\}_{l=1}^L$ encapsulates the parameters of the network comprised of L layers and $\Phi : \mathbb{R}^{w+1} \mapsto \mathbb{R}^h$ defines the mapping from the input space to the output space. The optimal parameters of the mapping function \mathcal{W}^* are computed by optimizing over the loss curve using gradient descent. Typically, Mean Squared Error (MSE) is used as a loss function, and hence, the optimization problem for regression can be mathematically stated as:

$$\mathcal{W}^* = \arg \min_{\mathcal{W}} \frac{1}{h} \sum_{i=1}^h \|Y_{t+i} - \Phi([x_t, \dots, x_{t-w}]; \mathcal{W})\|_2^2 \quad (3)$$

$$\mathcal{W}^* = \arg \min_{\mathcal{W}} \frac{1}{h} \sum_{i=1}^h \|Y_{t+i} - \hat{Y}_{t+i}\|_2^2 \quad (4)$$

where Y_{t+i} and \hat{Y}_{t+i} denotes the ground truth and the predicted value at time $t + i$ respectively.

4 Knowledge Forcing Framework

Figure 1 shows the overall architecture of the knowledge forcing framework. The input sequence is first passed to KDS, which comes up with forecasts according to the rules defined in its knowledge base. The input sequence is also passed on to the LSTM encoder, which computes a fixed latent representation, v of the input sequence. This encoded representation given by the last hidden state of the LSTM is used to initialize the internal state of the decoder. Instead of sequentially using the output of the decoder at the previous time step as an input for the next time step, we utilize predictions given by KDS as input to the decoder. These KDS predictions are also added to the output of the decoder, making a residual connection. This enables the knowledge forcing framework to encapsulate information from both KDS and LSTM in a complementary manner. In the following subsections, we further elaborate on the constituent KDS and LSTM architecture, along with the knowledge fusion mechanism.

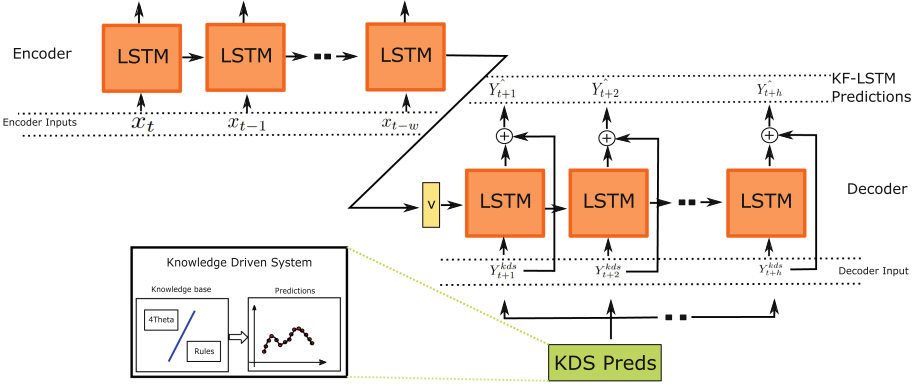


Fig. 1. KF-LSTM framework architecture.

4.1 Constituent KDS and LSTM Models

Knowledge forcing framework is agnostic to the underlying KDS model, however, for evaluations we utilize statistical methods 4Theta [23]. 4Theta is based on the original Theta [1] method, which models time series by decomposing it into theta lines. The theta lines are obtained by utilizing a parameter θ , which controls the curvature of theta lines. $0 < \theta < 1$ leads to less fluctuating lines, modeling the long term linear dependencies in the time series and higher-order $\theta > 1$ models the fluctuation, modeling the short-term attributes of the time series. For simplicity, we mathematically present a theta estimator using two theta lines in Eq. 5

$$Y_t = \omega_{\theta_1} Y_t^{\theta_1} + \omega_{\theta_2} Y_t^{\theta_2} \quad (5)$$

where ω_{θ_1} and ω_{θ_2} are the weights of the two theta lines. θ_1 and θ_2 model the long- and short-term characteristics of the original data, respectively. $Y_t^{\theta_1}$ represents the theta line at point t and can be obtained by the following equation Eq. 6

$$Y_t^\theta = \theta Y_t'' = \theta(Y_t - 2Y_{t-1} + Y_{t+2}) = \theta Y_t + (1 - \theta)(b + at) \quad (6)$$

where Y'' is the second difference of the data and b and a are the intercept and slope of simple linear regression in time Y^0 .

We employ a single layered LSTM model with 64 hidden units as our data-driven model. LSTM employs gating mechanism namely: input i_t , forget f_t , and output o_t gate which determines which long-term information to store and which short-term memory is to be read from the memory cell. This allows LSTMs to retain key information in the input sequence while ignoring less important parts. These long- and short-term information are preserved using internal state vectors C_t and h_t respectively. This can be represented mathematically by

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C'_t, h_t = o_t \cdot \Phi(C_t) \quad (7)$$

where Φ represents tangent function i.e., \tanh .

4.2 Knowledge Fused Optimization

The input \mathbf{X}_w is first given to KDS that makes an inference about future h values based on information explicitly defined in its knowledge base. The same input is also passed to the encoder of the KF-LSTM that encodes long- and short-term information in the historical data in its latent representations. We represent this vector containing latent representations as v , which consists of vector C_t and h_t from Eq. 7. The main difference of KF-LSTM from vanilla LSTM is in the decoder, In KF-LSTM, the KDS predictions are given as input to the decoder instead of giving the output of previous time stamp as an input to the decoder of the next time stamp. In addition to this, the predictions given by KDS are also added to the output given by the decoder, This makes a residual connection, where the decoder is connected in the residual connection whereas the KDS is connected via the skip connection. This changes the objective function learned by the decoder. The decoder now learns the residual function instead of learning input sequence to output sequence mapping. The residual function basically offsets any information that is either missing or incorrect in KDS predictions. This not only allows information to flow from KDS but enables KF-LSTM to correct inaccuracies in KDS predictions by utilizing hidden information contained in the historical data. As a result, KF-LSTM combines information in KDS and in historical data in a complementary and constructive way, where inaccuracies are suppressed and corrected. Let \mathbf{Y}_h^{LSTM} is the output of the LSTM decoder. Mathematically, this is calculated by

$$\mathbf{Y}_h^{LSTM} = \Phi(p(Y_t|v, Y_h^{KDS})) \quad (8)$$

where Y_h^{KDS} are the predictions given by the KDS. Since the final output of KF-LSTM is a summation of \mathbf{Y}_h^{LSTM} and Y_h^{KDS} . The optimization Eq. 4 can now be written as

$$\begin{aligned} \mathcal{W}^* &= \arg \min_{\mathcal{W}} \sum_{x \in \mathcal{Y}} \|\mathcal{Y} - (Y_h^{KDS} + Y_h^{LSTM})\|_2^2 \\ &= \arg \min_{\mathcal{W}} \sum_{x \in \mathcal{Y}} \|(\mathcal{Y} - Y_h^{KDS}) - Y_h^{LSTM}\|_2^2 \\ &= \arg \min_{\mathcal{W}} \sum_{x \in \mathcal{Y}} \|\xi_{KDS} - Y_h^{LSTM}\|_2^2 \end{aligned} \quad (9)$$

where \mathcal{Y} is the ground truth and ξ_{KDS} represents error in KDS predictions. As evident from the Eq. 9, KF-LSTM modifies the objective learned by the underlying LSTM, which is to minimize the error contained in KDS predictions. This is done by information contained in the historical data, which is encoded in vector v and is passed to the decoder and is used to initialize the internal states of the decoder. As a result, the overall KF-LSTM framework tries to combine the best of both world, which is not the case in other hybrid schemes.

5 Experiments and Results

5.1 Datasets

We evaluate knowledge forcing framework on 4 benchmark datasets belonging to different real-world applications. The datasets utilized are as follows: (1) *PeMSD7(M)* [31] dataset contains vehicular traffic information of District 7 of California containing data of 228 sensors from May to June 2012, (2) *Nasdaq* [21] dataset contains stock price (NASDAQ 100 index) information of 81 corporations, recorded every minute for 105 days, (3) *Energy* [3] dataset is made up of 26 different attributes related to energy consumption of different appliances in a single household, (4) *ETTM2* [28] datasets contains readings of electrical transformers like load and oil temperature, recorded every 15 min from July 2016 to July 2016.

5.2 Baseline Methods

We compare knowledge forcing framework against more than 8 baseline methods including recent transformer based methods: ETSformer [27], Autoformer [28], Spatial-Temporal Transformer Networks (STTN) [30], Informer [33], Reformer [11], LogTrans [14], graph-based networks: Graph-Wavenet [20], Spatio-Temporal Graph Convolutional (STGCN) [31], Convolutional and Recurrent Neural Networks: LSTM [8], Diffusion Convolutional Recurrent Neural Network (DCRNN) [15], Long- and Short-term time series network (LSTNet) [12], Multi-level Construal Neural Network (MLCNN) [5], Neural basis expansion (N-BEATS) [19]. However, we only report the results claimed by the authors of individual methods and do not reproduce results except for ETSformer [27], Autoformer [28] and Nbeats [19].

Table 1. Results of KF-LSTM framework along with baseline methods on all the datasets. A lower MSE and MAE value represents better forecasts. Best results are written with green, while second best are written with blue color.

Methods		KF-LSTM		ETSformer [27]		Autoformer [28]		Informer [33]		LSTnet [12]		Nbeats [19]		MLCNN [5]		STTN [30]	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PeMSD7(M)	3	13.26	1.96	21.34	2.78	19.27	2.72	-	-	-	-	23.04	3.71	-	-	16.32	2.14
	6	25.17	2.63	28.52	3.24	31.36	3.41	-	-	-	-	40.20	4.54	-	-	28.84	2.70
	9	19.94	2.37	25.10	2.91	32.26	3.21	-	-	-	-	64.80	5.22	-	-	36.60	3.03
Nasdaq	3	0.011	0.029	2.22	0.780	1.85	0.45	-	-	0.134	0.093	10.18	1.11	0.133	0.091	-	-
	6	0.022	0.041	3.03	0.830	3.50	0.53	-	-	0.272	0.135	10.76	1.13	0.266	0.130	-	-
	12	0.052	0.061	3.06	0.850	1.93	0.46	-	-	0.569	0.195	6.864	0.78	0.546	0.186	-	-
Energy	3	174.0	2.70	209.38	2.42	259.85	3.86	-	-	240.56	1.82	236.85	2.76	228.92	1.88	-	-
	6	188.40	1.43	234.09	2.82	289.00	4.00	-	-	249.64	2.39	258.57	3.02	255.68	2.38	-	-
	12	251.30	2.03	262.44	3.32	314.71	4.35	-	-	285.27	3.11	291.38	3.12	281.57	3.04	-	-
ETTM2	96	0.187	0.301	0.189	0.28	0.255	0.34	0.365	0.45	3.142	1.37	-	-	-	-	-	-
	192	0.251	0.355	0.253	0.32	0.281	0.34	0.533	0.56	31.54	1.37	-	-	-	-	-	-
	336	0.746	0.582	0.314	0.36	0.339	0.37	1.363	0.89	3.160	1.37	-	-	-	-	-	-

5.3 Results

Table 1 shows results obtained by knowledge forced LSTM along with baseline methods. Knowledge forced LSTM consistently achieves SotA results on all the datasets except ETTm2, where it achieves second-best result on MAE metric but still manages SotA result on MSE metric except for the horizon of 336. Averaging across all the prediction lengths, knowledge forced LSTM achieves **23%** reduction in MSE and **22%** reduction in MAE compared to previous SotA on PeMSD7(M) dataset. **91%** reduction in MSE and **68%** reduction in MAE on Nasdaq dataset and **14%** and **10%** reduction on MSE and MAE on Energy dataset. On ETTm2 dataset, knowledge forced LSTM achieves **9%** reduction in MSE for horizon 96 and 192, while it achieves second-best results on MAE metric except for the forecasting horizon of 336, where knowledge forced LSTM achieves third-best result overall. We believe that this is due to an extremely long forecasting horizon, which a shallow LSTM was unable to model properly. Nevertheless, average across all the dataset and evaluation metrics, knowledge forced LSTM achieves **20%** improvement over the previous SotA.

Figure 2 shows plots of forecasts made by KF-LSTM and recent transformer-based methods, Autoformer [28] and ETSformer [27]. KF-LSTM not only follows the trend more accurately, but also models subtle variations and extremas more accurately, which is of real importance in domains such as finance.

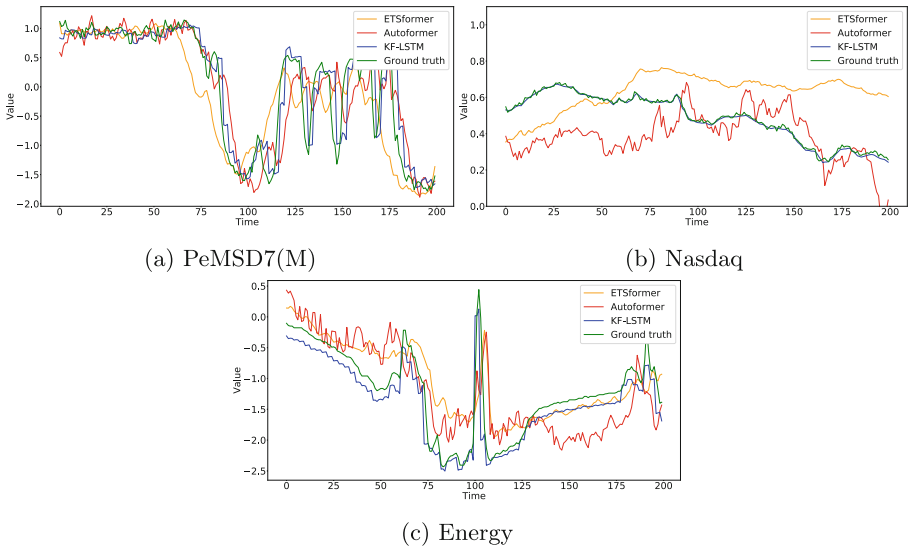


Fig. 2. Prediction of KF-LSTM, Autoformer and ETSformer on PeMSd7(M), Nasdaq, and energy dataset for horizon 3

5.4 Ablation Study

Table 2. Comparison of MSE and MAE metrics of vanilla LSTM and KDS without the fusion mechanism for PeMSD7(M) dataset. Percentage improvement given by the KF-LSTM over the vanilla LSTM and KDS is also given.

Metric (Horizon)	Vanilla LSTM	KDS	KF-LSTM
MSE (3/ 6/ 9)	14.44/ 30.7/ 35.4	15.9/ 30.7/ 47.5	13.26/ 25.17/ 19.94
Percentage Improvement in MSE (3/ 6/ 9)	8%/ 18%/ 44%	16%/ 18%/ 58%	-
MAE (3/ 6/ 9)	2.24/ 2.73/ 3.50	2.13/ 2.90/ 3.44	1.96/ 2.63/ 2.37
Percentage Improvement in MAE (3/ 6/ 9)	13%/ 4%/ 32%	8%/ 9%/ 31%	-

In this section, we study the impact of knowledge forcing by evaluating the underlying KDS and LSTM models in isolation without the fusion mechanism. Table 2 shows the result of LSTM and KDS model along with the results obtained by KF-LSTM. Additional rows highlighting percentage improvement over constituent LSTM and KDS models are also included. As evident from the Table 2, employing the proposed knowledge forcing mechanism improves the overall accuracy, with improvements as high as **58%** over constituent knowledge and data domains.

Moreover, Table 2 also highlights the ability of knowledge forcing mechanism to dynamically adapt based on the information contained within respective modalities since percentage improvement is not constant for each of the constituent domain, but infact varies according to the efficacy of each domain.

5.5 KDS Model Independence

Table 3. Results of KF-LSTM framework with different KDS model on PeMSD7(M) dataset for forecasting horizon of 3

KDS model	MSE	MAE
Rule-based KDS	13.36	2.04
Statistical method as KDS	13.26	1.96

In this section, we verify the agnostic nature of the knowledge forcing mechanism towards the underlying KDS model. This is important because the knowledge

base of KDS can take different forms like rule-based relations, mathematical formulations etc., and it is desirable that the knowledge fusion framework is flexible enough to incorporate different KDS models. For this, we change the underlying KDS from a statistical based method to a rule-based model. The rules for forecasting are made from the scheme proposed in [4], which borrows concepts from graph-based network by considering each time series observations as a node of the graph and calculating relations between the nodes with correlation functions. Table 3 shows the results of knowledge forcing with a rule-based KDS model.

5.6 Comparison with Ensemble Methods

In this section, we evaluate the knowledge forced LSTM framework against the ensemble technique, that is a commonly used method in the literature for combining predictions of two or more models. Table 4 shows the comparison of KF-LSTM with the ensemble method. For every dataset, KF-LSTM gives substantially superior performance compared to the ensemble. This is primarily due to the error correction capabilities of KF-LSTM that in a way mitigate some of the inaccuracies in the final output.

Table 4. Comparison of ensemble methods with KF-LSTM on all the datasets.

Dataset	Horizon	MSE		MAE	
		Ensemble	KF-LSTM	Ensemble	KF-LSTM
PeMSD7(M)	3	13.70	13.26	2.04	1.96
	6	24.90	25.17	2.64	2.63
	9	30.01	19.94	3.00	2.37
Nasdaq	3	0.032	0.011	0.052	0.029
	6	0.035	0.022	0.054	0.041
	12	0.107	0.052	0.086	0.061
Energy	3	208.83	174.00	2.75	2.70
	6	245.50	188.40	1.76	1.43
	12	272.58	251.30	2.40	2.03
ETT	96	0.32	0.19	0.31	0.30
	192	0.35	0.25	0.36	0.36
	336	0.76	0.75	0.60	0.58

6 Conclusion

In this paper, we present a novel hybrid framework, KF-LSTM, that combines KDS approaches with LSTM in a way, where not only useful information contained in the constituent domains are integrated but their inaccuracies and shortcomings are also catered for in the final output. We evaluate KF-LSTM against

recent SotA baseline methods on 4 time series benchmark forecasting datasets. Despite being a relatively shallow network, KF-LSTM outperforms recent transformer and graph-based models handsomely and establishes new SotA on almost every dataset. This highlights the effectiveness of the proposed fusion framework. We also show that KF-LSTM is flexible towards different KDS models. This will prove useful in applicability of KF-LSTM in real-world applications, where KDS may comprise of different and diverse knowledge bases. We also compare KF-LSTM with current ensemble based hybrid schemes. KF-LSTM significantly outperforms ensemble technique in terms of overall accuracy. The ability of KF-LSTM to constructively utilize knowledge and data domain will prove useful in unlocking the true potential of artificial intelligence especially in critical applications where domain knowledge is also crucial.

References

1. Assimakopoulos, V., Nikolopoulos, K.: The theta model: a decomposition approach to forecasting. *Int. J. Forecast.* **16**(4), 521–530 (2000)
2. Box, G.E., Jenkins, G.M., Reinsel, G.: Time series analysis: forecasting and control Holden-day San Francisco. *BoxTime Series Analysis: Forecasting and Control Holden Day 1970* (1970)
3. Candanedo, L.M., Feldheim, V., Deramaix, D.: Data driven prediction models of energy use of appliances in a low-energy house. *Energy Build.* **140**, 81–97 (2017)
4. Chattha, M.A., van Elst, L., Malik, M.I., Dengel, A., Ahmed, S.: KENN: enhancing deep neural networks by leveraging knowledge for time series forecasting. arXiv preprint [arXiv:2202.03903](https://arxiv.org/abs/2202.03903) (2022)
5. Cheng, J., Huang, K., Zheng, Z.: Towards better forecasting by fusing near and distant future visions. In: *AAAI*, pp. 3593–3600 (2020)
6. Han, H., et al.: STGCN: a spatial-temporal aware graph learning method for poi recommendation. In: *2020 IEEE International Conference on Data Mining (ICDM)*, pp. 1052–1057. IEEE (2020)
7. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint [arXiv:1503.02531](https://arxiv.org/abs/1503.02531) (2015)
8. Hochreiter, S., Schmidhuber, J.: LSTM can solve hard long time lag problems. In: *Advances in Neural Information Processing Systems*, pp. 473–479 (1997)
9. Hunter, J.S.: The exponentially weighted moving average. *J. Qual. Technol.* **18**(4), 203–210 (1986)
10. Kaushik, S., et al.: Ai in healthcare: time-series forecasting using statistical, neural, and ensemble architectures. *Front. Big Data* **3**, 4 (2020)
11. Kitaev, N., Kaiser, L., Levskaya, A.: Reformer: the efficient transformer. arXiv preprint [arXiv:2001.04451](https://arxiv.org/abs/2001.04451) (2020)
12. Lai, G., Chang, W.C., Yang, Y., Liu, H.: Modeling long-and short-term temporal patterns with deep neural networks. In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 95–104 (2018)
13. Larrea, M., Porto, A., Irigoyen, E., Barragán, A.J., Andújar, J.M.: Extreme learning machine ensemble model for time series forecasting boosted by PSO: application to an electric consumption problem. *Neurocomputing* **452**, 465–472 (2021)
14. Li, S., et al.: Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Adv. Neural. Inf. Process. Syst.* **32**, 5243–5253 (2019)

15. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network. In: Data-driven Traffic Forecasting, ICLR 2018 Conference, pp. 1–16 (2017)
16. Makridakis, S., Hibon, M.: The M3-competition: results, conclusions and implications. *Int. J. Forecast.* **16**(4), 451–476 (2000)
17. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: The m4 competition: results, findings, conclusion and way forward. *Int. J. Forecast.* **34**, 802–808 (2018)
18. Moghram, I., Rahman, S.: Analysis and evaluation of five short-term load forecasting techniques. *IEEE Trans. Power Syst.* **4**(4), 1484–1491 (1989)
19. Oreshkin, B.N., Carпов, D., Chapados, N., Bengio, Y.: N-beats: neural basis expansion analysis for interpretable time series forecasting. arXiv preprint [arXiv:1905.10437](https://arxiv.org/abs/1905.10437) (2019)
20. Pan, Z., Liang, Y., Wang, W., Yu, Y., Zheng, Y., Zhang, J.: Urban traffic prediction from spatio-temporal data using deep meta learning. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1720–1730 (2019)
21. Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., Cottrell, G.: A dual-stage attention-based recurrent neural network for time series prediction. arXiv preprint [arXiv:1704.02971](https://arxiv.org/abs/1704.02971) (2017)
22. Smyl, S.: A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *Int. J. Forecast.* **36**(1), 75–85 (2020)
23. Spiliotis, E., Makridakis, S., Assimakopoulos, V.: The m4 competition in progress. In: 38th International Symposium on Forecasting (2018)
24. Towell, G.G., Shavlik, J.W.: Knowledge-based artificial neural networks. *Artif. Intell.* **70**(1–2), 119–165 (1994)
25. Tran, S.N., Garcez, A.S.d.: Deep logic networks: inserting and extracting knowledge from deep belief networks. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(2), 246–258 (2018)
26. Wang, W.: Some fundamental issues in ensemble methods. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 2243–2250. IEEE (2008)
27. Woo, G., Liu, C., Sahoo, D., Kumar, A., Hoi, S.: ETSformer: exponential smoothing transformers for time-series forecasting. arXiv preprint [arXiv:2202.01381](https://arxiv.org/abs/2202.01381) (2022)
28. Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In: Advances in Neural Information Processing Systems, vol. 34 (2021)
29. Xie, Q., Hovy, E., Luong, M.T., Le, Q.V.: Self-training with noisy student improves Imagenet classification. arXiv preprint [arXiv:1911.04252](https://arxiv.org/abs/1911.04252) (2019)
30. Xu, M., Dai, W., Liu, C., Gao, X., Lin, W., Qi, G.J., Xiong, H.: Spatial-temporal transformer networks for traffic flow forecasting. arXiv preprint [arXiv:2001.02908](https://arxiv.org/abs/2001.02908) (2020)
31. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. arXiv preprint [arXiv:1709.04875](https://arxiv.org/abs/1709.04875) (2017)
32. Zhang, R., Ashuri, B., Deng, Y.: A novel method for forecasting time series based on fuzzy logic and visibility graph. *Adv. Data Anal. Classif.* **11**(4), 759–783 (2017)
33. Zhou, H., et al.: Informer: beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of AAAI (2021)