

Chapter 6

Self-Attention-Driven DL for accurate electricity forecasting in smart building environments

This chapter focuses on the fourth contribution of this thesis. We provide an introduction and key contribution to the work in Section 6.1. The description of the mechanism is given in Section 6.2. Section 6.3 demonstrates the proposed work for an efficient LTEF. The experimental setup and result are given in Section 6.4. Section 6.5 covers the summary obtained using the proposed technique.

6.1 Introduction

With accelerating urbanization and technological advancement, EC in SBs has increased significantly. These buildings, equipped with advanced technologies, interconnected systems, and data-driven tools, aim to optimize energy efficiency, enhance occupant comfort, and reduce environmental impact. However, as urban energy demand grows, managing consumption in these intelligent spaces becomes increasingly complex. Buildings account for around 40–50% of global energy usage, highlighting

the urgent need for smarter, sustainable solutions. SBs use tools like smart meters, occupancy sensors, and Building Management Systems (BMS) to monitor real-time energy patterns. Yet, energy behavior in such environments is highly dynamic and non-linear, influenced by variables like occupancy, temperature, humidity, solar radiation, and HVAC performance. Traditional models like ARIMA struggle with such complexity, prompting the shift toward ML and DL. DL models, particularly those with sequence learning capabilities, better capture temporal patterns but often require large labeled DSs. This study introduces a novel DL framework with a softmax-based self-attention (SA) mechanism that prioritizes critical features across multivariate time series inputs. By accurately forecasting energy use based on conditions like temperature, solar input, and occupancy, the model enables real-time adjustments in HVAC and lighting systems. This approach improves energy efficiency, reduces operational costs, and supports sustainability goals in smart building environments.

Key contributions of this study include:

- The use of a Temporal Convolutional Network (TCN)-based DL model for energy consumption forecasting in SBs.
- Evaluation of the model’s LT prediction capability.
- Comparative performance analysis using publicly available DSs and metrics like MAE, RMSE, and R^2 score.
- Integration of a SA mechanism to identify critical features, leading to improved prediction accuracy over conventional DL models.

6.2 Theoretical Background

This section provides a detailed explanation of the problem and outlines the proposed methodology used to address it.

6.2.1 Attention Mechanism

The SA mechanism is a technique that allows a model to weigh the importance of different parts of the input sequence when encoding each element. It is widely used in Transformer architectures and time series modeling.

1. Input Representation

Given an input sequence represented as a matrix $X \in \mathbb{R}^{n \times d}$, where n is the number of elements (e.g., time steps or tokens) and d is the embedding dimension, we project the input into three learned representations:

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V$$

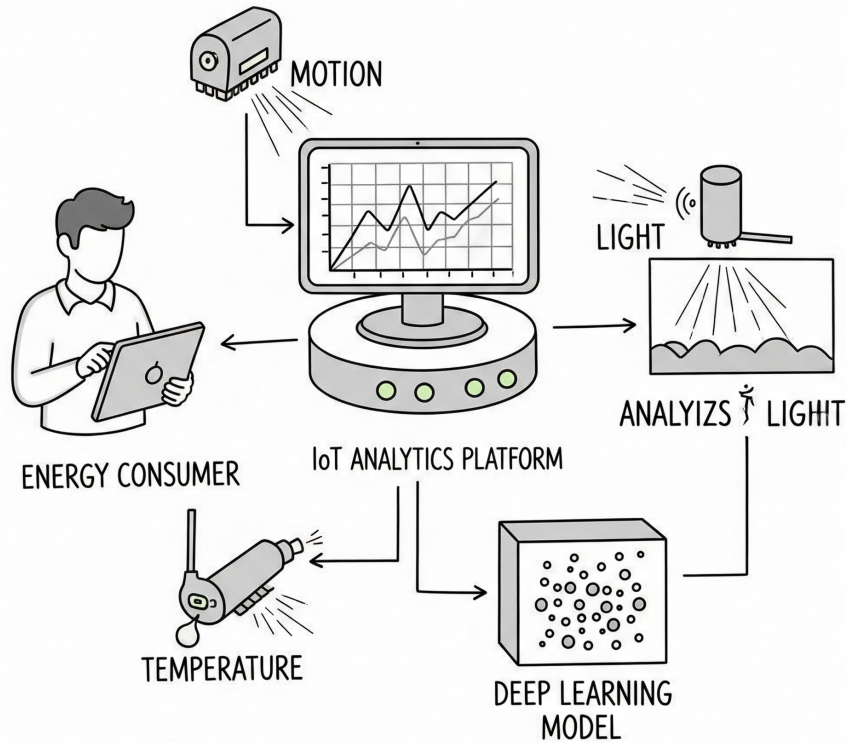
Here, $W^Q, W^K, W^V \in \mathbb{R}^{d \times d_k}$ are learnable weight matrices, and $Q, K, V \in \mathbb{R}^{n \times d_k}$ represent the Query, Key, and Value matrices respectively.

2. Scaled Dot-Product Attention

To compute the attention scores between queries and keys, we use the dot product followed by scaling and a softmax operation:

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V$$

- QK^T : computes the similarity between queries and keys.
- $\sqrt{d_k}$: scales the scores to avoid large gradients.
- softmax: normalizes scores into probabilities.
- The result is a weighted sum of the values V , where the weights reflect the relevance of each input element.



(a)

FIGURE 6.1: Architectural overview of proposed system.

Recent research has turned to DL, which has shown strong potential to improve forecasting accuracy through better feature representation and sequence modeling. In this work, we propose a hybrid model called TCN-SA-GRU that integrates TCN, SA mechanisms, and GRUs to create a more robust and reliable predictive system.

6.2.2 Model Overview

Figure 6.1 illustrates the four-tier architecture of our proposed framework for forecasting energy consumption (EC) in smart residential settings. Each tier has a specific role, working together to turn raw sensor readings into actionable insights for residents and facility managers.

1. Data Collection Tier Dataset 1 (D1) – Multimodal data gathered over 12 months from a 200 m² office space in Vienna. It records tenant presence, occupant behaviour, and both indoor and outdoor environmental conditions.

Dataset 2 (D2) – Time-series data captured in a 2015 smart office building in Berkeley, California [92]. IoT sensors attached to office equipment log temperature, humidity, solar radiation, and other parameters every 1–15 minutes. All data are pre-collected and ready for analysis.

6.3 Proposed Methodology

Energy consumption forecasting plays a crucial role in smart EM and efforts to cut down greenhouse gas emissions. It’s all about staying one step ahead early warnings from accurate predictions help building managers optimize energy use, keep occupants comfortable, reduce operating costs, and improve overall productivity. These forecasts also make buildings more resilient, especially when tracking energy data at regular intervals typically every 15 minutes—thanks to real-time sensors.

To improve forecasting accuracy, our approach builds on earlier hybrid models and tailors them to fit the unique patterns found in EC data. The goal is to extract and use meaningful, high-level information to make better predictions.

We introduce a new hybrid model called TCN-SA-GRU 6.2, which combines TCN for spotting patterns in energy data with (GRUs for tracking LT trends. We’ve also added a SA layer to help the model focus on the most important time points, especially when energy usage patterns repeat. This layered design lets the model learn more effectively and deliver smarter, more reliable predictions.

6.4 Experimental Setup and Result Discussion

In comparison to the suggested hybrid model, this section explains the DSs, data pretreatment, performance measures, and alternative DL models. The models are

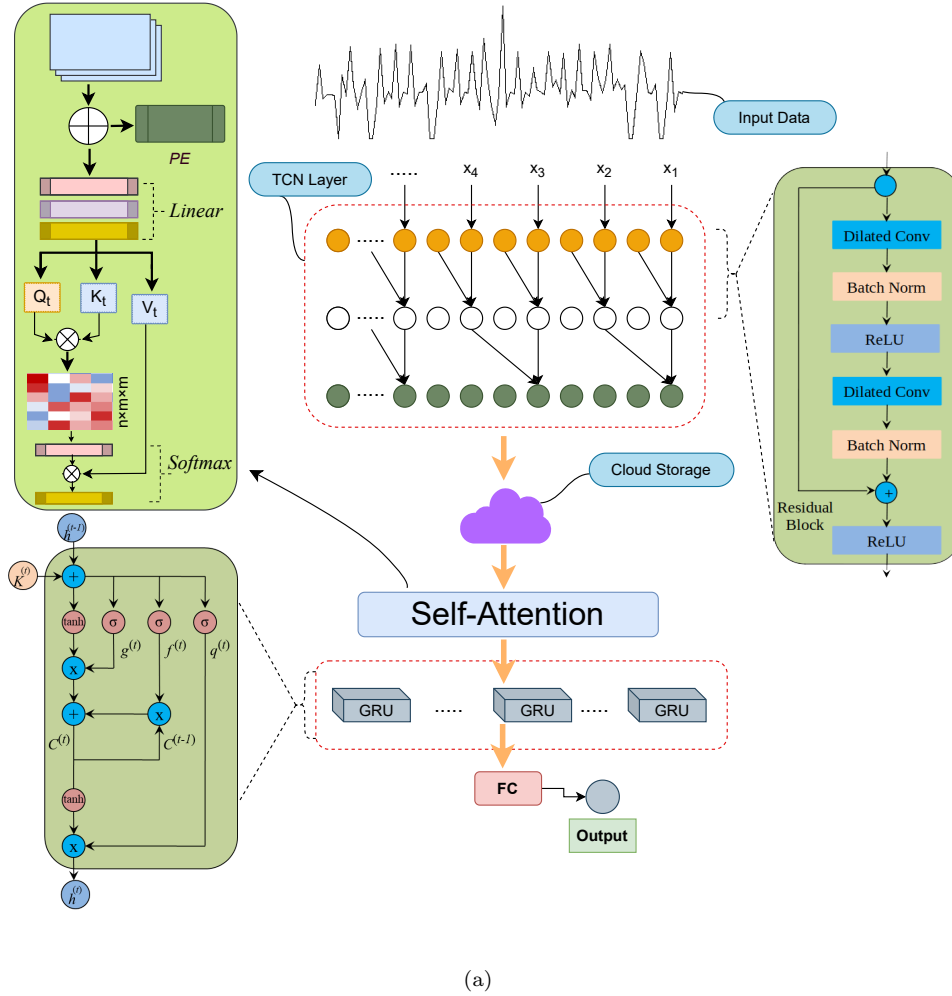


FIGURE 6.2: Proposed hybrid dl model with SA for EC prediction

compared on different-2 window sizes, such as 4, 12, 20, and 28, with different time stamps in hr i.e. 1, 3, 5, and 7. We have taken the best of each model concerning window size. The model’s documentation contains Activation Function (AF), Loss (L), Optimizer (O), Dense Layer (DL), Learning Rate (LR), and Batch Size (BS).

Table I 6.1 lists the several models developed in this research based on algorithms like CNN, LSTM, Bi-LSTM, CNN-LSTM, and TCN as well as suggested models like TCN-LSTM, TCN-GRU, and TCN-Bi-LSTM, TCN-SA-GRU and their parameters.

TABLE 6.1: Experimental Models with their parameters

Experimental Models	Parameters
LR	Shape of Input: lag x 3 dimension
ARIMA	Lag p=5, difference d=0, Past Error q=5
RNN	RNN layer (75, AF = Sigmoid, Loss = MAE, O = Adam)
LSTM	LSTM layer (Nodes = 75, AF = ReLU, Loss= MAE ,O = Adam)
Bi-LSTM	Bi-LSTM layer (50, Dropout: 0.2) x1, DL (1), Loss: MSE, O = Adam, LR: 0.001, BS: 128
CNN-LSTM	CNN Filters = 2, Flatten Layer, LSTM, Nodes = 100, AF = ReLU, DL (1), Loss = MAE, O = Adam
TCN	TCN layer (Filter = 32, AF = Relu, Kernel size = 5, dilation = [1,2,4,8,16,32] x1), Dropout = 0.2, DL (1), Loss = MSE, O = Adam, BS = 32, LR = 0.001)
TCN-RNN	TCN layer (Filter = 32, AF = Relu, Kernel size = 5, dilation = [1,2,4,8,16,32] x1), Dropout = 0.2, DL (1), Loss = MSE, O = Adam, BS = 32, LR = 0.001), RNN layer (100, AF = Sigmoid, Loss = MAE, O = Adam)
TCN-LSTM	TCN layer (Filter = 32, AF = Relu, Kernel size = 5, dilation = [1,2,4,8,16,32] x1), Dropout = 0.2, DL (1), Loss = MSE, O = Adam, Batch size = 32, LR = 0.001), LSTM layer (50, Dropout: 0.2) x1, DL (1), Loss = MSE, O = Adam, LR = 0.001, BS = 128
TCN Bi-LSTM	TCN layer (Filter = 32, AF = Relu, Kernel size = 5, dilation = [1,2,4,8,16,32] x1), Dropout = 0.2, DL (1), Loss = MSE, O = Adam, BS = 32, LR = 0.001), Bi-LSTM layer (50, Dropout: 0.2) x1, DL (1), Loss = MSE, O = Adam, LR = 0.001, BS = 128
TCN-GRU	TCN layer (Filter = 32, AF = Relu, Kernel size = 5, dilation = [1,2,4,8,16,32] x1), Dropout = 0.2, DL (1), Loss = MSE, O = Adam, Batch size = 32, LR = 0.001), GRU layer (50, Dropout: 0.2) x1, DL (1), Loss = MSE, O = Adam, LR = 0.001, BS = 128
TCN-SA-GRU	TCN layer (Filter = 32, AF = Relu, Kernel size = 5, dilation = [1,2,4,8,16,32] x1), Dropout = 0.2, DL (1), Loss = MSE, O = Adam, Batch size = 32, LR = 0.001), SA layer (input_dim = 64, hidden_dim = 64, num_heads = 4, dropout_rate = 0.2, O = Adam, LR = 0.001), GRU layer (50, Dropout: 0.2) x1, DL (1), Loss = MSE, O = Adam, LR = 0.001, BS = 128

6.4.1 Results Evaluation

This study presents a hybrid deep learning (HDL) model, TCN-SA-GRU (TAG), designed to forecast EC in SBs by leveraging historical data from IoT sensors. The model is evaluated in the dataset and the results are in table 6.2 :

In the TAG model:

Multivariate input features are first processed through a GRU layer, which captures temporal dependencies in both forward and backward directions.

The output is then passed to a SA mechanism, which enhances the model’s ability to identify and focus on the most relevant temporal features.

TABLE 6.2: Performance comparison of all forecasting models using MAE, RMSE, and R^2 metrics across varying time lags (1, 3, 5, and 7 hours). Bold values represent the best results for each metric at each time horizon.

Error Metric	Models	1-hour	3-hour	5-hour	7-hour
MAE	LR	6.2261	7.9251	7.1650	6.8673
	ARIMA	7.1421	8.2405	6.0124	6.3866
	RNN	6.8102	6.4168	7.1042	7.8611
	LSTM	5.4261	4.4275	6.8854	5.7426
	Bi-LSTM	5.3951	4.2010	5.3172	5.0597
	CNN-LSTM	4.8210	5.4224	6.7844	6.8260
	TCN	3.6617	5.2041	5.0045	4.0978
	TCN-RNN	6.9426	5.7436	7.8228	5.1330
	TCN-LSTM	3.1385	4.3182	4.6686	3.6136
	TCN Bi-LSTM	3.8158	3.8862	4.9676	3.2016
	TCN-SA-GRU (D1)	3.6820	3.3020	3.0410	2.8840
	TCN-SA-GRU (D2)	3.5520	3.4520	3.2410	2.8000
RMSE	LR	8.3625	9.7793	10.1792	8.1210
	ARIMA	10.0074	11.3610	9.2574	8.4705
	RNN	8.4345	7.4022	9.9498	9.1481
	LSTM	6.8601	6.6709	8.5607	6.3900
	Bi-LSTM	7.2083	5.9610	7.4296	7.0124
	CNN-LSTM	7.6434	7.8437	9.6364	8.4135
	TCN	5.2610	6.5183	6.8465	6.3301
	TCN-RNN	7.4875	7.8851	9.1351	7.5585
	TCN-LSTM	5.3082	5.9533	6.0205	5.2585
	TCN Bi-LSTM	5.0179	5.3207	5.2260	4.2051
	TCN-SA-GRU (D1)	5.1120	4.7510	4.4410	3.8740
	TCN-SA-GRU (D2)	4.4020	4.7240	4.5720	3.8210
R^2 Score	LR	0.8730	0.8500	0.8320	0.8920
	ARIMA	0.8210	0.7520	0.8590	0.8860
	RNN	0.8870	0.9260	0.8360	0.8610
	LSTM	0.9300	0.9110	0.8730	0.9280
	Bi-LSTM	0.9170	0.9420	0.9150	0.8850
	CNN-LSTM	0.9210	0.9060	0.8520	0.8840
	TCN	0.9460	0.9240	0.9220	0.9340
	TCN-RNN	0.9280	0.9170	0.8600	0.9260
	TCN-LSTM	0.9430	0.9500	0.9370	0.9410
	TCN Bi-LSTM	0.9480	0.9510	0.9470	0.9680
	TCN-SA-GRU (D1)	0.9590	0.9660	0.9720	0.9810
	TCN-SA-GRU (D2)	0.9500	0.9620	0.9725	0.9830

The attention enhanced features are forwarded to a global server, where the complete TCN-SA-GRU model is trained. This global model is later deployed to local servers for real-time EC prediction.

The model’s performance was tested against several baseline algorithms, including LR, LSTM, Bi-LSTM, TCN, CNN-LSTM, and TCN-RNN. The experiments were conducted on a system with an Intel Core i7-10785H CPU (2.60 GHz) and 16 GB RAM. The DSs were divided into 90% training and 10% testing, with original minute-level data resampled into 15-minute intervals. Multiple lag features were added to enhance learning accuracy.

6.4.1.1 Performance Evaluation

The proposed TAG model demonstrated superior predictive accuracy compared to all baseline models. At the 7-hour forecast horizon, it achieved a MAE of 2.884 and RMSE of 3.874 on Dataset 1 (D1), and MAE of 2.8 and RMSE of 3.821 on Dataset 2 (D2). Additionally, it recorded the highest R^2 values, with 0.981 for D1 and 0.983 for D2, confirming its strong reliability and precision in energy consumption forecasting.

6.4.1.2 Visualization and Analysis

Figure 6.3 illustrates the comparison between actual and predicted EC for both DSs. Subfigure (a) shows results for D1 (California), and (b) for D2 (Vienna). The x-axis represents time in hours, and the y-axis shows load in kilowatts (kW). The close alignment between actual and predicted values highlights the model’s accuracy.

In summary, the TCN-SA-GRU model demonstrates strong potential for real-time EC forecasting in SBs, providing both precision and stability across various temporal resolutions and operational conditions.

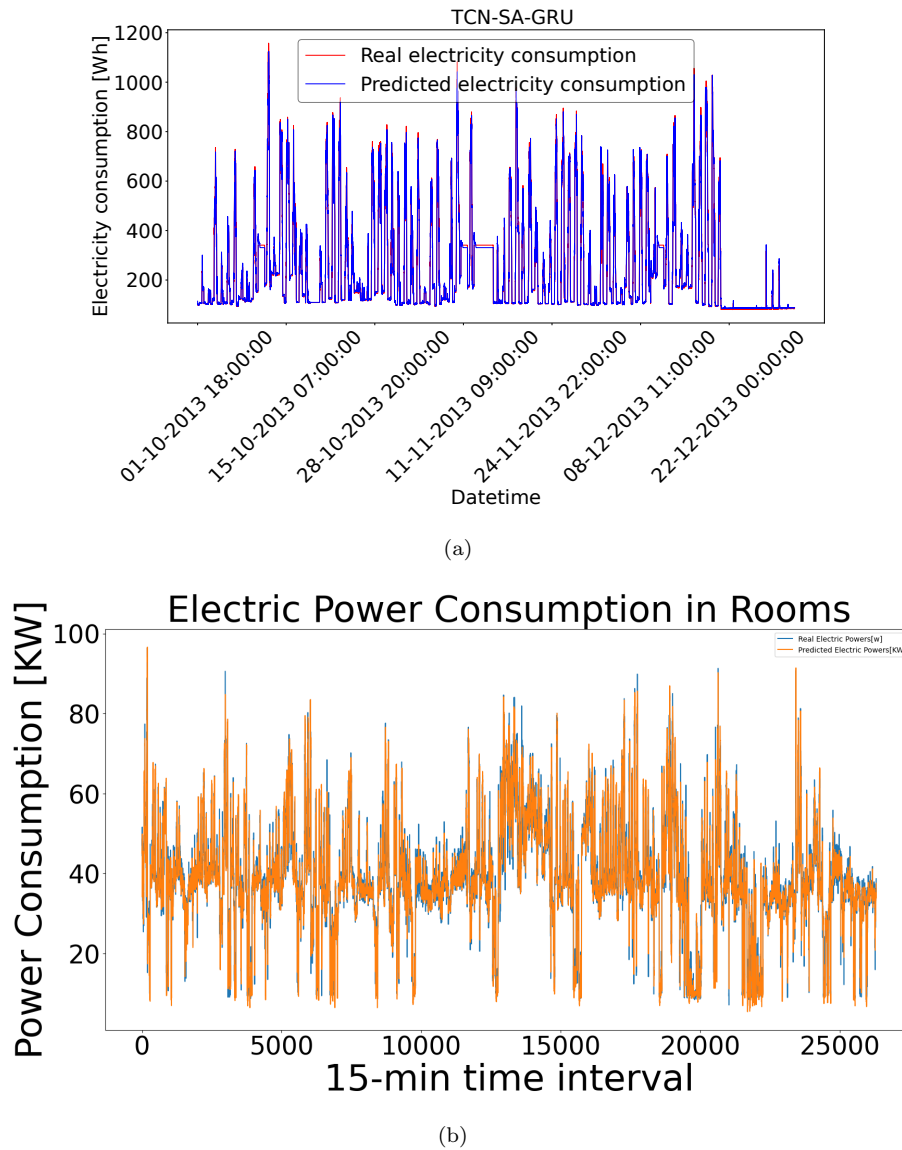


FIGURE 6.3: (

a) Actual v/s predicted graph using TCN-SA-GRU on univariant TS data and (b) Actual v/s predicted graph using TCN-SA-GRU on multivariant TS data.

6.5 Summary

This research addresses the challenge of making precise predictions for energy consumption (EC) in SBs, whose usage patterns are extremely dynamic and driven by factors such as occupant behavior and new devices. A hybrid deep model named TCN-SA-GRU is proposed to tackle this, by integrating TCN for feature extraction, SA for contextuality, and GRUs for LTD capture. Experimental results indicate

that the integrated approach performs better than the conventional models in terms of prediction accuracy. Future research will seek to apply the model to various geographic locations and climates by integrating local environmental conditions like temperature, humidity, and daylight.