

DOI: 10.37943/16YIKA8050

Samat Kabdygali

Master of Science, Department of Science and Innovations
samat.kabdygali@alumni.nu.edu.kz, orcid.org/0009-0006-8804-7461
Astana IT University, Kazakhstan

Ruslan Omirgaliyev

Master of Science, Senior-lecturer, Department of Computer Engineering
Ruslan.omirgaliyev@astanait.edu.kz, orcid.org/0000-0003-3834-2359
Astana IT University, Kazakhstan

Timur Tursynbayev

PhD in Power supply and electrification, Executive Director
timur.tursynbayev@gmail.com, orcid.org/0009-0002-5857-1343
ALE “Kazakhstan Association of Digital Energy”, Kazakhstan

Korhan Kayisli

PhD, Assoc.professor, Faculty of Engineering, Department of Electrical-
Electronics Engineering
korhankayisli@gazi.edu.tr, orcid.org/0000-0001-8456-1478
Gazi University, Turkey

Nurkhat Zhakiyev*

PhD, Department of Science and Innovations
nurkhat.zhakiyev@astanait.edu.kz, orcid.org/0000-0002-4904-2047
Astana IT University, Kazakhstan

DEEP RECURRENT NEURAL NETWORKS IN ENERGY DEMAND FORECASTING: A CASE STUDY OF KAZAKHSTAN'S ELECTRICAL CONSUMPTION

Abstract: The critical transformation of the energy sector demands innovative approaches to ensure the reliability and efficiency of energy systems. In this pursuit, this study delved into the potential of Deep Recurrent Neural Networks (DRNNs) for forecasting energy demand, using a comprehensive dataset detailing Kazakhstan's electrical consumption over a span of two years. Traditional statistical models have historically played a role in energy demand prediction, but the growing intricacy of the energy landscape calls for more advanced solutions. The paper presented a comparison of the DRNN with other traditional and machine learning models and highlighted the superior performance of DRNNs, especially in capturing complex temporal relationships.

The energy sector is confronting unprecedented challenges due to population growth and the integration of diverse energy sources, leading to increased demand and system strains. Accurate energy demand prediction is essential for system reliability. Traditional models, though widely used, often overlook intricate variables like weather patterns and temporal factors. Through rigorous methodology, encompassing exploratory data analysis, feature engineering, and hyperparameter optimization, an optimized DRNN model was developed. The results demonstrated the DRNN's exceptional capability in processing complex time-series data, as evidenced by its attainment of an R-squared value of 83.6%. Additionally, it achieved Mean Absolute Errors and Root Mean Squared Errors of less than 2%. However, there were noticeable

deviations in some predictions, suggesting areas for refinement. This research underscores the significance of DRNNs in energy demand prediction, highlighting their advantages over traditional models while also noting the need for ongoing optimization. The findings underscore DRNN's promise as a robust forecasting tool, pivotal for the energy sector's future resilience and efficiency.

Keywords: recurrent neural networks; Kazakhstan; electrical consumption; forecasting system.

Introduction (Literature review)

The energy sector is undergoing a significant transformation catalyzed by the surge in population growth and the integration of diversified energy sources. These factors have led to an increased strain on energy systems, creating supply uncertainties and risks of blackouts. In this evolving landscape, accurate forecasting of energy demand has become paramount for effective system management and planning. Traditional statistical models have been commonly used for this purpose but often fail to account for complex variables like weather conditions, time of day, and holidays. More recently, machine learning (ML) techniques have been adopted to address these shortcomings.

This literature review aims to present an overview of various approaches to energy demand prediction, emphasizing the application of machine learning models such as recurrent neural networks (RNN), convolutional neural networks (CNN), and support vector regression (SVR) [1,2]. To overcome the limitations of conventional models, researchers have started exploring the application of machine learning techniques for energy demand prediction. Among various ML techniques, RNN, CNN, MLP, and SVR have emerged as front-runners in the sector. Agrawal et al. [1] leveraged Long Short-Term Memory (LSTM) RNN models for long-term load forecasting, achieving a Mean Absolute Percentage Error (MAPE) of 6.54%. Their study utilized a public dataset spanning twelve years, showcasing the model's high accuracy. Conversely, Taheri et al. [2] compared different deep learning algorithms for long-term energy consumption prediction. Their findings showed that Deep-RNN (DRNN) outperformed gradient boosting (GB) and support vector machines (SVM), with monthly average errors being lower for DRNN. Deep Recurrent Neural Networks (DRNNs) are a specialized class of neural networks optimized for sequence prediction problems. Unlike traditional RNNs, which usually consist of a single layer of recurrent connections, DRNNs employ multiple layers. This enables DRNNs to capture higher-order dependencies and complexities, making them particularly well-suited for tasks like energy demand prediction, which involve intricate relationships among multiple variables. Energy demand prediction is not a trivial task; it encompasses a variety of variables such as weather conditions, holidays, time of day, and day of the week. Traditional statistical models often fail to account for these complexities. Taheri et al.'s study [2] delves into this aspect, presenting DRNNs as a more effective solution for long-term energy consumption prediction. One of the unique strengths of DRNNs are their ability to handle sequence data efficiently. Traditional machine learning models often disregard the temporal sequence of data points, which can be a critical factor in energy demand prediction. DRNNs maintain a memory; of past sequences, enabling them to capture temporal dependencies effectively. This makes them ideally suited for time-series data, a common data type in energy systems. Training DRNNs require significant computational power and time, which could be a limitation for real-time applications. Also, like other deep learning models, DRNNs may require large datasets for training to prevent overfitting [3]. Research works [4,5] have highlighted that DRNNs are highly effective for applications in sequentially collected data, outperforming other machine learning algorithms in accuracy. Despite their capabilities, DRNNs are computationally intensive, especially when multiple layers and neurons are involved. DRNNs represent a promising approach in the

domain of energy demand prediction, overcoming several limitations of traditional statistical models and other machine learning algorithms. With their deep architecture and ability to capture complex temporal relationships, they offer a robust model for accurate, long-term forecasting in energy systems, albeit with some computational challenges. Therefore, their application in the energy sector should be the subject of continuous research and optimization to fully realize their potential.

Another practical usage of DRNN is image segmentation (for example, cardiac). In [6] authors reviewed 60 works by using deep learning methods in cardiac image segmentation. They did comparative analysis of different network architectures with accuracies. Paper [7] proposes to use neural network algorithms for detecting heart sound signal. Their algorithms include lots of input parameters and computationally expensive for running. In the result part, authors achieved 97.2% accuracy. Deep learning algorithms (namely ResNet) can be effectively used for Inferior Vena Cava (IVC) filter segmentations [8]. Authors analyzed 84 CT scans and by applying 3D-CNN and Swin-UNETR architecture retrieved IVC filters.

Research paper [9] contains information about different variations of LSTM (long short-term memory) for recognizing sequential data, text, video and audio.

Moreover, deep neural network models can also be used in transportation sector. For example, authors of [10] paper tested deep neural network framework on complex road geometry to predict dynamics of driver-vehicle system.

Several studies [11,12] have examined the performance of sequential data using LSTM, GRU, and Transformer models. In terms of model training speed [11], GRU is 29.29% faster than LSTM in processing the same dataset; and in terms of performance, GRU outperforms LSTM in the long text and small dataset scenario and is inferior to LSTM in other scenarios. Considering two aspects, performance and cost of computing power, the performance cost ratio of GRU is higher than that of LSTM, which is 23.45%, 27.69% and 26.95% higher in terms of precision, recall and F1 respectively.

Two metrics, the BLEU score and the ROUGE score, are utilized to estimate the performance of the models [12]. The BLEU-4 score is 0.386, 0.402, and 0.482 for the RNN+LSTM, RNN+GRU, and Transformer models respectively. The precision, recall, and F1 score studies for the ROUGE Score show similar results to those of the BLEU Score training. Both evaluation metrics show that the Transformer model outperforms both RNN variants.

In addition to Recurrent Neural Networks (RNNs), models based on the CNN-LSTM architecture have demonstrated high performance in forecasting energy demand [13,14]. Specifically, research papers [14,15] highlight the application of the CNN-LSTM framework across various datasets, resulting in precise predictions of energy consumption with an accuracy rate ranging between 94% and 96%. This indicates the robustness of the CNN-LSTM model in handling the complexities of energy demand forecasting [16,17].

Recent advances in Transformer models have significantly expanded the field of time series analysis, particularly in energy forecasting tasks. The scientific papers listed as references [18,19] provide a comprehensive exploration of the application of transformer models in this context. These models are characterized by their ability to skillfully manage and accurately describe the intricate relationships inherent in complex data sets. This is particularly important in the field of energy forecasting, where the dynamism and complexity of the data require sophisticated analytical approaches. The Transformer architecture, with its advanced mechanisms for handling sequential data, provides a robust framework for capturing temporal dependencies and nuances, thereby enhancing the accuracy and reliability of predictive analysis in the energy sector [20].

In conclusion, the energy sector's growing complexity and dynamic nature necessitate advanced forecasting methods, and machine learning techniques, especially deep learning mod-

els like DRNNs, have proven superior in addressing this need. While these models offer enhanced accuracy and the ability to process complex temporal data, they also pose challenges in terms of computational intensity and data requirements. The continued exploration and optimization of these technologies, including DRNNs, CNN-LSTM, and Transformer models, are crucial for improving long-term energy demand prediction and application in other sectors like healthcare and transportation.

Methods and Materials

Deep Recurrent Neural Networks (RNNs) are a type of neural network architecture designed to handle sequential data by incorporating multiple layers of recurrent units. They are an extension of the standard RNN architecture, which has the limitation of struggling to capture long-range dependencies in sequential data. Deep RNNs aim to address this issue by stacking multiple recurrent layers on top of each other. Here's a general overview of the architecture:

1. **Recurrent Layers:** The core building blocks of a Deep RNN are the recurrent layers. These layers maintain hidden states that capture information from the input sequence at different time steps. In a deep architecture, you have multiple recurrent layers stacked on top of each other.
2. **Time Steps:** Each layer in the deep RNN processes the input sequence one time step at a time. The output from one time step becomes the input to the next, effectively passing information forward through the network.
3. **Hidden States:** The hidden state at each time step in a recurrent layer is a vector that represents the network's memory of the past inputs. In a deep RNN, each layer has its own set of hidden states. The output of one layer's hidden states becomes the input for the next layer.
4. **Multiple Layers:** Deep RNNs typically consist of multiple recurrent layers, which allows them to capture complex dependencies in the data. The first layer processes the raw input, and subsequent layers process the output of the previous layer.
5. **Non-Linear Activation Functions:** Similar to other neural networks, deep RNNs often use non-linear activation functions, such as the hyperbolic tangent (tanh) or Rectified Linear Unit (ReLU), to introduce non-linearity into the network and enable it to learn complex patterns.
6. **Backpropagation Through Time (BPTT):** Training a deep RNN involves backpropagating the error through time to adjust the network's parameters, including the weights and biases in each layer. This process is known as Backpropagation Through Time (BPTT).

Deep RNNs have been used in a wide range of applications, including natural language processing, speech recognition, and time series analysis. They can capture intricate patterns in sequential data, making them a valuable tool for tasks that involve understanding and generating sequences.

It's important to note that deep RNNs can be challenging to train due to issues like vanishing and exploding gradients. To mitigate these problems, variations of RNNs, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks, are often used in practice. These variants have mechanisms to better control the flow of information through the network's hidden states and have become the standard choice for deep recurrent architectures.

The conventional approach for constructing deep RNNs is remarkably straightforward: we arrange multiple RNN layers in a stacked fashion. When dealing with a sequence of length "T" the initial RNN layer generates an output sequence of the same length "T," which then serves as the input for the subsequent RNN layer. In this concise section, we demonstrate this architectural pattern and provide a simple example of how to implement such stacked RNNs. Illustrated in Figure 1 below is a deep RNN with "L" hidden layers. Each hidden state processes

a sequential input and yields a sequential output. Additionally, it's important to note that any RNN cell (represented as a white box in Figure 1) at each time step relies on both the previous time step's value within the same layer and the value of the previous layer at the corresponding time step.

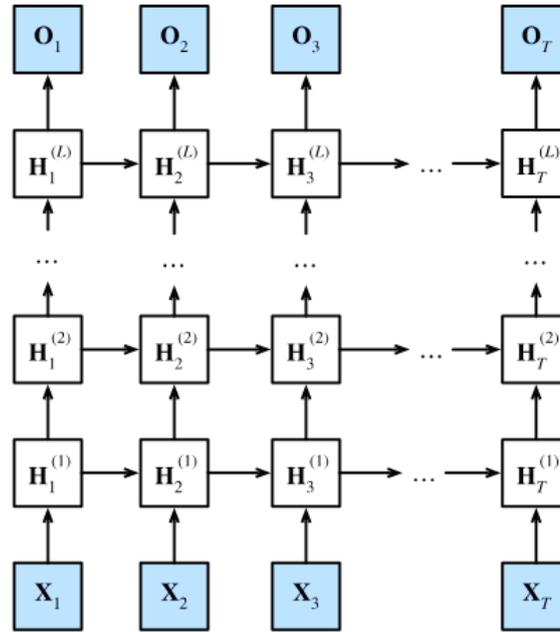


Figure 1. Architecture of DRNN Model

A Deep Recurrent Neural Network (RNN) can be mathematically formulated as a series of equations representing how information flows through the network. Here is the mathematical formulation of a basic deep RNN with “L” hidden layers:

Notation:

- t represents the time step.
- $X(t)$ is the input at time step t
- $H_1^{(l)}$ is the hidden state of the l -th layer at time step t
- $O(t)$ is the output at time step t

Input to Hidden State of Layer 1 can be described by using the following equation:

$$H_1^{(t)} = f(W^{(1)} * X(t) + U^{(1)} * H_1^{(t-1)}) \quad (1)$$

where $f()$ is the activation function (e.g., tanh or ReLU);

$W^{(1)}$ – the weight matrix for the input;

$U^{(1)}$ – the weight matrix for the previous hidden state of the first layer;

$H_1^{(t-1)}$ – the hidden state of the first layer at the previous time step.

Calculation of next hidden layers ($2 \leq l \leq N$):

$$H_l^{(t)} = f(W^{(l)} * H_{l-1}^{(t)} + U^{(l)} * H_l^{(t-1)}) \quad (2)$$

where $W^{(l)}$ is the weight matrix for the hidden state of the previous layer;

$U^{(l)}$ – the weight matrix for the previous hidden state of the same layer;

$H_{l-1}^{(t)}$ – the hidden state of the previous layer at the current time step;

$H_l^{(t-1)}$ – the hidden state of the current layer at the previous time step.

Output at time step t is computed as follows:

$$O(t) = f(V * H_L^{(t)}) \quad (3)$$

where V is the weight matrix for the output layer.

$H_L^{(t)}$ – the hidden state of the final layer at the current time step.

The above equations represent the basic mathematical formulation of a deep RNN. In practice, it is possible to use more advanced RNN cell types like LSTM (Long Short-Term Memory) or GRU (Gated Recurrent Unit) cells and implement various optimization techniques to make training more efficient and stable. The choice of activation functions, the number of layers, and the architecture may vary based on the specific problem which we are solving.

In this paper, it was employed a Deep Recurrent Neural Network (DRNN) to analyze a detailed dataset capturing Kazakhstan's electrical consumption from April 2018 to April 2020. Each entry in the dataset presents consumption metrics for a specific hour and date. Covering a broad spectrum of zones and regions, the dataset provides a consolidated overview of Kazakhstan's electrical demand, further segmented into regions such as North, South, West, and East. Detailed data for specific locales, including Semey and Karaganda GRES, are also available. Moreover, the dataset encompasses secondary load metrics for selected regions. Given its extensive duration and hourly granularity, this dataset serves as a critical resource for evaluating electricity consumption trends, identifying regional energy imbalances, informing policy and infrastructure planning, and supporting predictive demand forecasting. The data primarily originates from national electricity boards or associated regulatory agencies.

Deep Recurrent Neural Network were developed using this dataset, the DRNN model includes pandas for structured data manipulation, numpy for numerical operations, matplotlib and seaborn for visualization, sklearn for preprocessing and metrics, tensorflow for deep learning operations, and optuna for hyperparameter optimization.

The feature engineering stage comes next. Here, lag features, which are historical data values, are introduced. Specifically, three lag features are generated to capture the power consumption data from the previous three time steps. Additionally, a rolling window statistic, a rolling mean in this case, is computed over a window of three time periods to offer a smoothed version of power consumption.

The entire dataset is then normalized to fall between the range 0 and 1 using the MinMax-Scaler. This normalization ensures stable and efficient training of neural networks. For data analysis, a correlation heatmap is plotted to visualize correlations between different features. Following this, pair plots are generated for select columns to visually inspect relationships between pairs of variables. The equation of the scaling is:

$$X_{scaled} = (X - X_{min}) / (X_{max} - X_{min}) \quad (4)$$

The dataset is prepared for modeling by dividing it into training, validation, and test subsets, which constitute 60%, 20%, and 20% of the total data, respectively. A utility function create dataset structures the data to make it suitable for time-series prediction using recurrent neural networks. The aim is to train the model to predict the data at time “t+1” based on the data available at time “t”.

Hyperparameter optimization is a critical step before model training. Optuna is employed to search for the best hyperparameters, including the type of RNN layer (SimpleRNN, LSTM, or GRU), the number of units in each RNN layer, dropout rates, learning rate, and the optimizer type (Adam, RMSprop, or SGD). The objective for Optuna’s optimization is the validation loss, with the aim to minimize it.

Once the best hyperparameters are identified, the deep learning model is constructed. This model is a deep recurrent neural network with three RNN layers followed by dropout layers for regularization and a dense layer for output. The model’s architecture is influenced by the best parameters from the optuna search. Furthermore, a learning rate scheduler is incorporated which reduces the learning rate by 10% after the initial 10 epochs to enhance the stability of training.

Post-training, the model’s performance is assessed using various metrics. Loss curves are plotted to visualize the model’s training and validation loss across epochs. The model’s predictive accuracy on the test set is then evaluated using the Root Mean Squared Error (RMSE), R-squared, and the Mean Absolute Error (MAE). To provide a visual comparison between the actual and predicted power consumption, plots are generated for two subsets of the test data: the first 100 and 1000 data points.

The equations of RMSE, R^2 , and MAE are:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (5)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \bar{y}_i)^2}{\sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (6)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (7)$$

Where, y_i is the actual value

\hat{y}_i is the predicted value

\bar{y}_i is the mean of the actual values

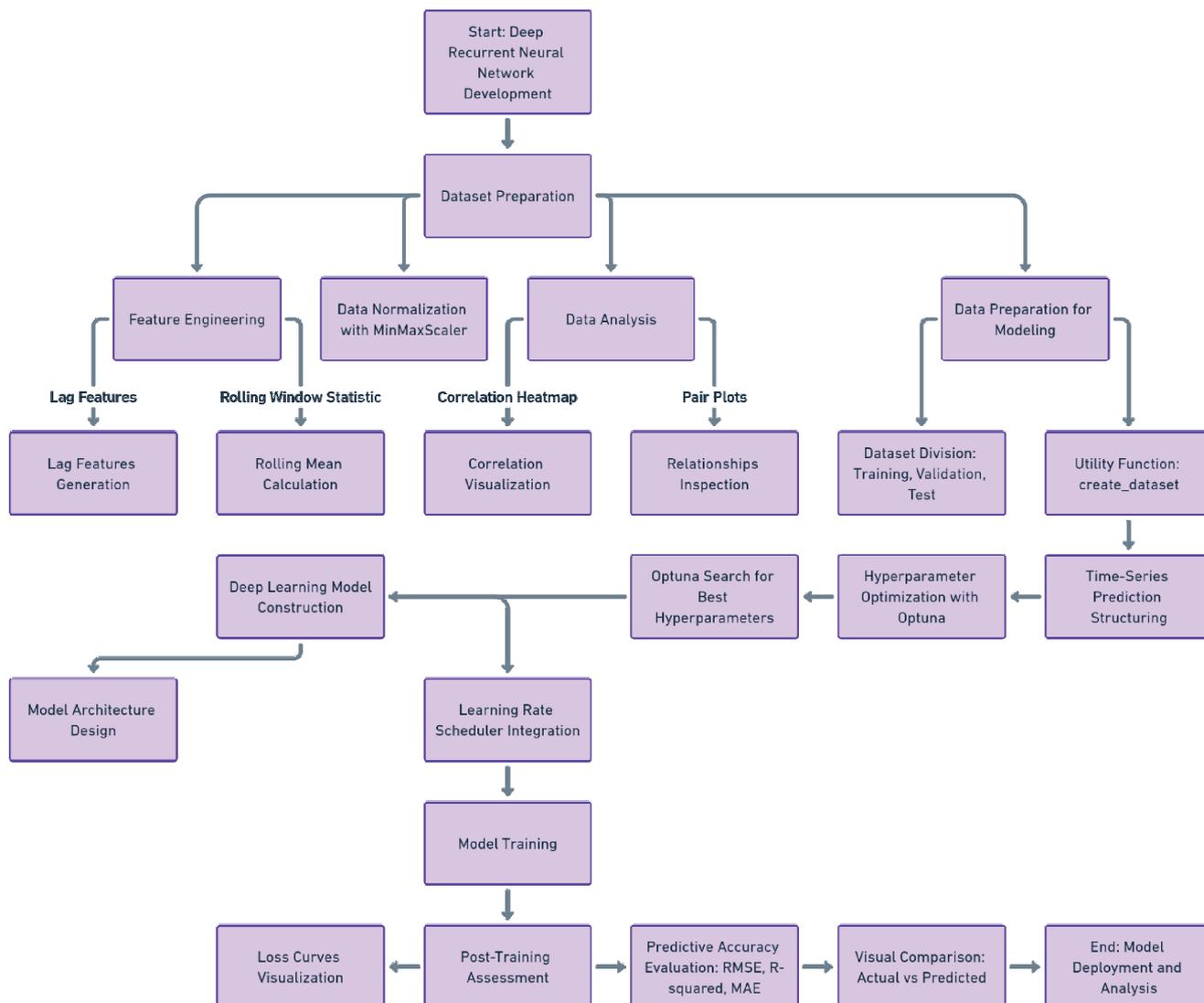


Figure 2. Deep Recurrent Neural Network Development Process Flow Chart

The flowchart shows the stages of deep recurrent neural network (RNN) development. As discussed above, it begins with data preparation, which includes feature engineering and normalization, followed by an analysis phase that examines the relationships in the data. Then, it moves to model preparation, which includes partitioning the dataset and creating a utility function. This is followed by model building, which involves optimizing the hyperparameters using tools such as Optuna and designing the model architecture. Model training involves tuning the learning rate, evaluating post-training results using loss visualization and accuracy metrics, and finally deploying the model and analytically comparing predicted and actual results.

Results and Discussion

Before training model some data were analyzed using correlation heat map and pair plots. Correlation heatmaps and pair plots are essential tools in exploratory data analysis, offering a visual summary of relationships between variables. Correlation heatmaps display the strength and direction of relationships through color-coding, enabling rapid identification of positively, negatively, or uncorrelated variables. Pair plots, on the other hand, provide both univari-

ate (through histograms or density plots on the diagonal) and bivariate insights (via scatter plots off the diagonal), revealing distributions and patterns of interaction between variables. Together, these tools facilitate a quick overview of large datasets, guiding analysts towards deeper investigation or highlighting areas for data cleaning. However, it's crucial to remember that correlation doesn't equate to causation. Figure 2 shows correlation between all features of dataset.

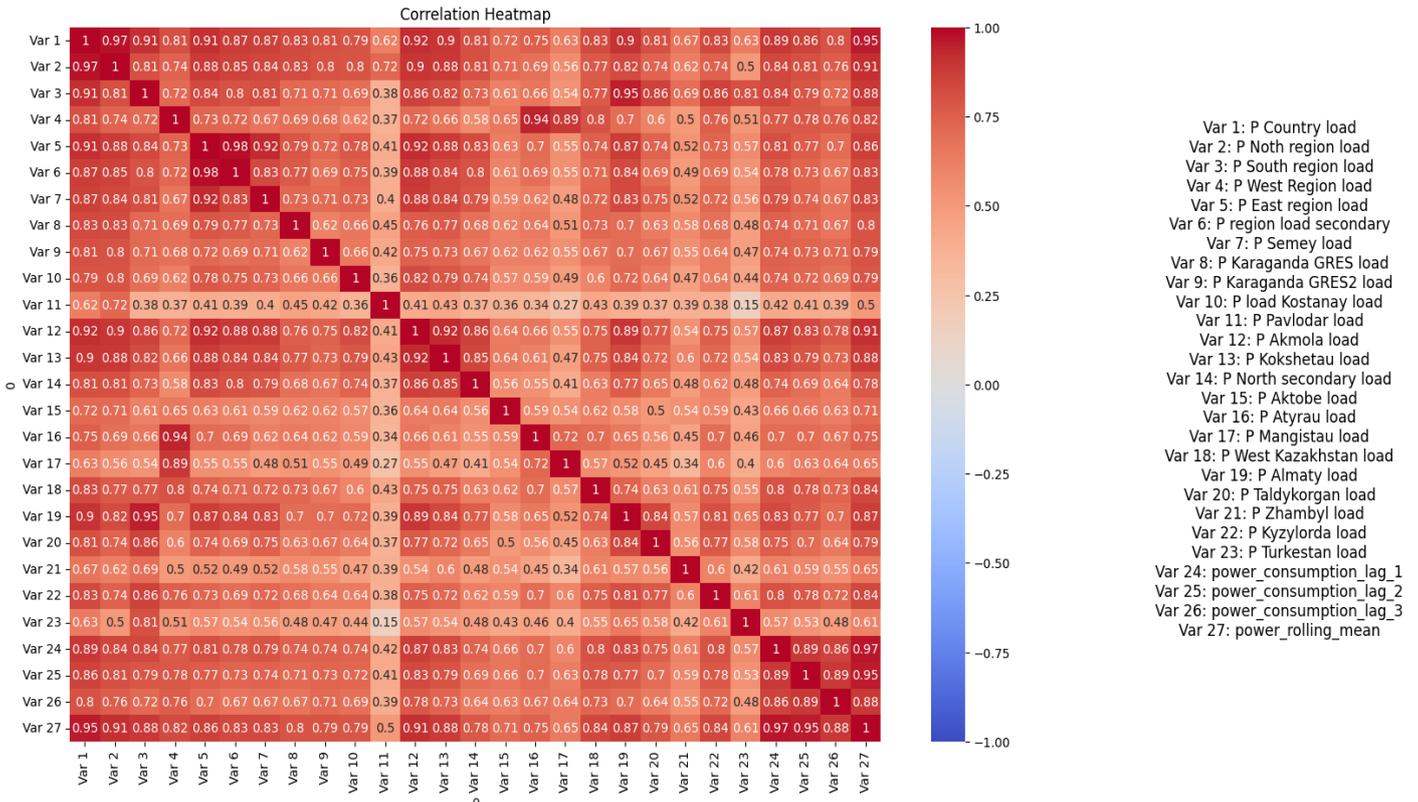


Figure 3. Correlation heat map

This correlation heatmap visually represents the relationships between various power consumption or load metrics across different regions. The color-coded matrix ranges from deep red, indicating strong positive correlations (values near 1), to deep blue, suggesting strong negative correlations (values near -1). The diagonal line from the top left to the bottom right signifies a perfect positive correlation, as any metric is always perfectly correlated with itself. Overall, the heatmap shows that many regions and metrics have strong positive correlations, suggesting that changes in one region or metric often coincide with similar changes in others, while a few areas display weak or negative correlations.

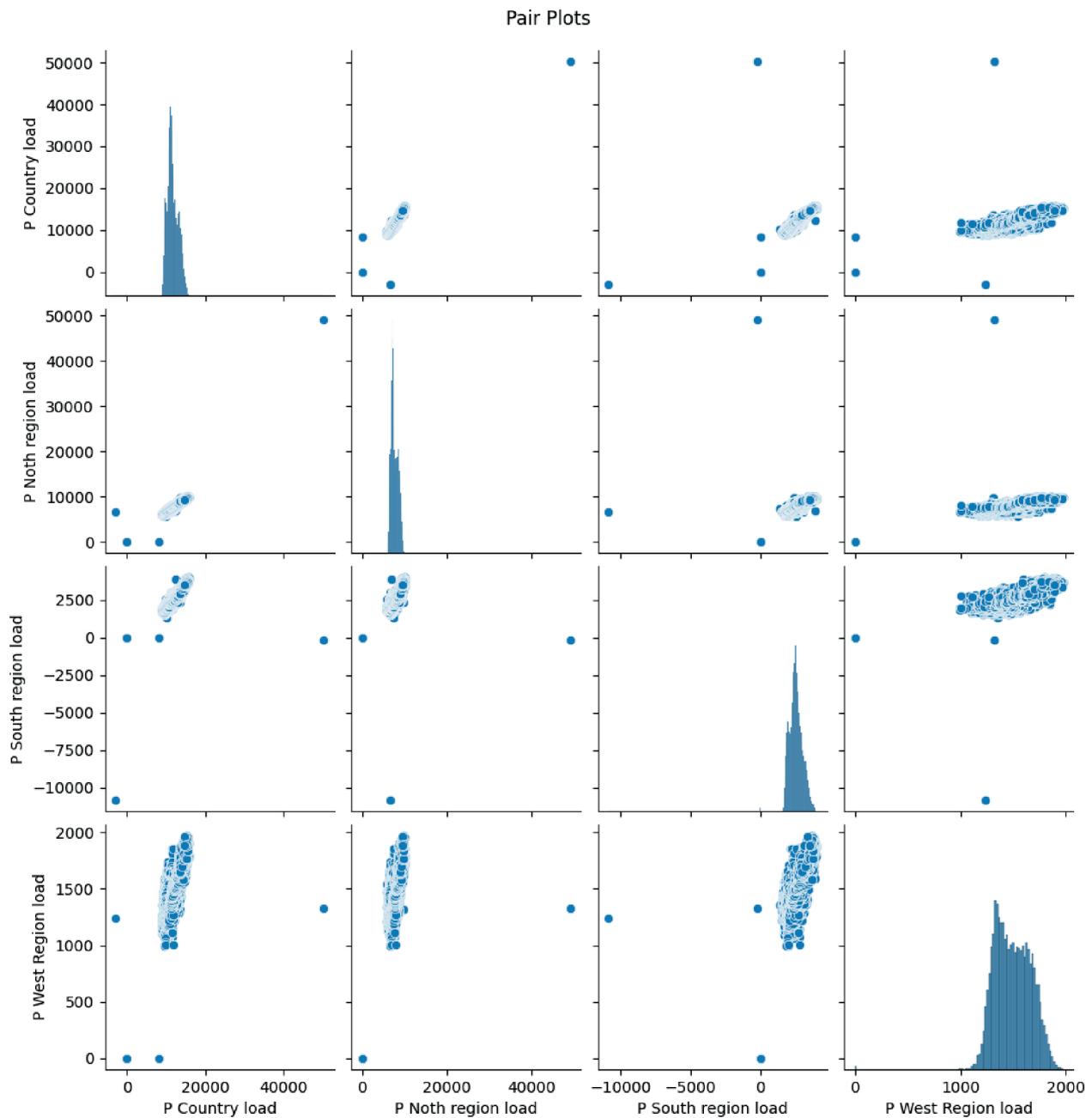


Figure 4. Pair Plots

The pair plot in Figure 3 provides a visual comparison of the relationships and distributions among four different power load metrics: 'P Country load', 'P North region load', 'P South region load', and 'P West Region load'. On the diagonal, the histograms show the distribution of each metric individually. The other scatter plots reveal the bivariate relationships between the metrics. For instance, the relationship between 'P Country load' and 'P North region load' exhibits a clustered pattern, implying specific common trends or groupings. Similarly, other plots also demonstrate various patterns or clusters of data points. Some of the metrics appear to have stronger linear relationships, while others show more dispersed scatter patterns, hinting at less direct correlations.

Based on this analysis, a Deep recurrent neural network (RNN) model were trained for a time series dataset. Initially, the dataset is split into training (60%), validation (20%), and testing (20%) sets. A helper function create dataset is defined to format the data for time

series forecasting with a specified look_back period. The data is then transformed using this function to produce input-output pairs. Subsequently, the model leverages the Optuna library for hyperparameter tuning: different RNN architectures (SimpleRNN, LSTM, GRU), number of units, dropout rates, learning rates, and optimizers are explored to determine the best model configuration. The objective of the tuning is to minimize the validation loss. After identifying the best parameters, the optimal RNN model is built and trained for 50 epochs, with a learning rate scheduler reducing the rate after 10 epochs. Finally, the training and validation loss is visualized over the epochs to assess the model's performance.

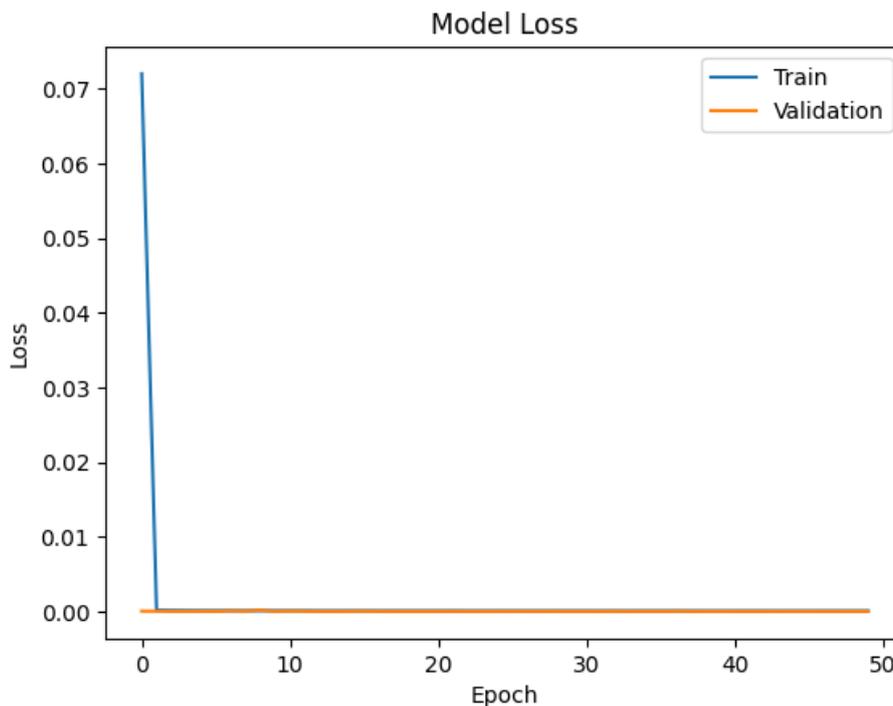


Figure 5. Loss function graph.

The provided graph in Figure 4 displays the model loss across epochs for both training and validation sets. Notably, the training loss sees a rapid decrease in the initial epochs and then stabilizes, suggesting a swift learning process. Meanwhile, the validation loss remains consistently low, hinting at effective model generalization. Although there's a discernible gap between the training and validation losses, with the former being lower, it's not alarmingly wide. Thus, overfitting doesn't seem to be a predominant issue, indicating that the model appears to be performing proficiently on both datasets.

Also, the model exhibits a fitting performance with an R-squared value of 83.6%, while the RMSE of 0.0126 and MAE of 0.0056 indicate the model's error magnitude, which should be assessed relative to the data's scale, distribution, and the specific application context.

Finally, Actual vs Predicted plots were designed to visualize the model performance:

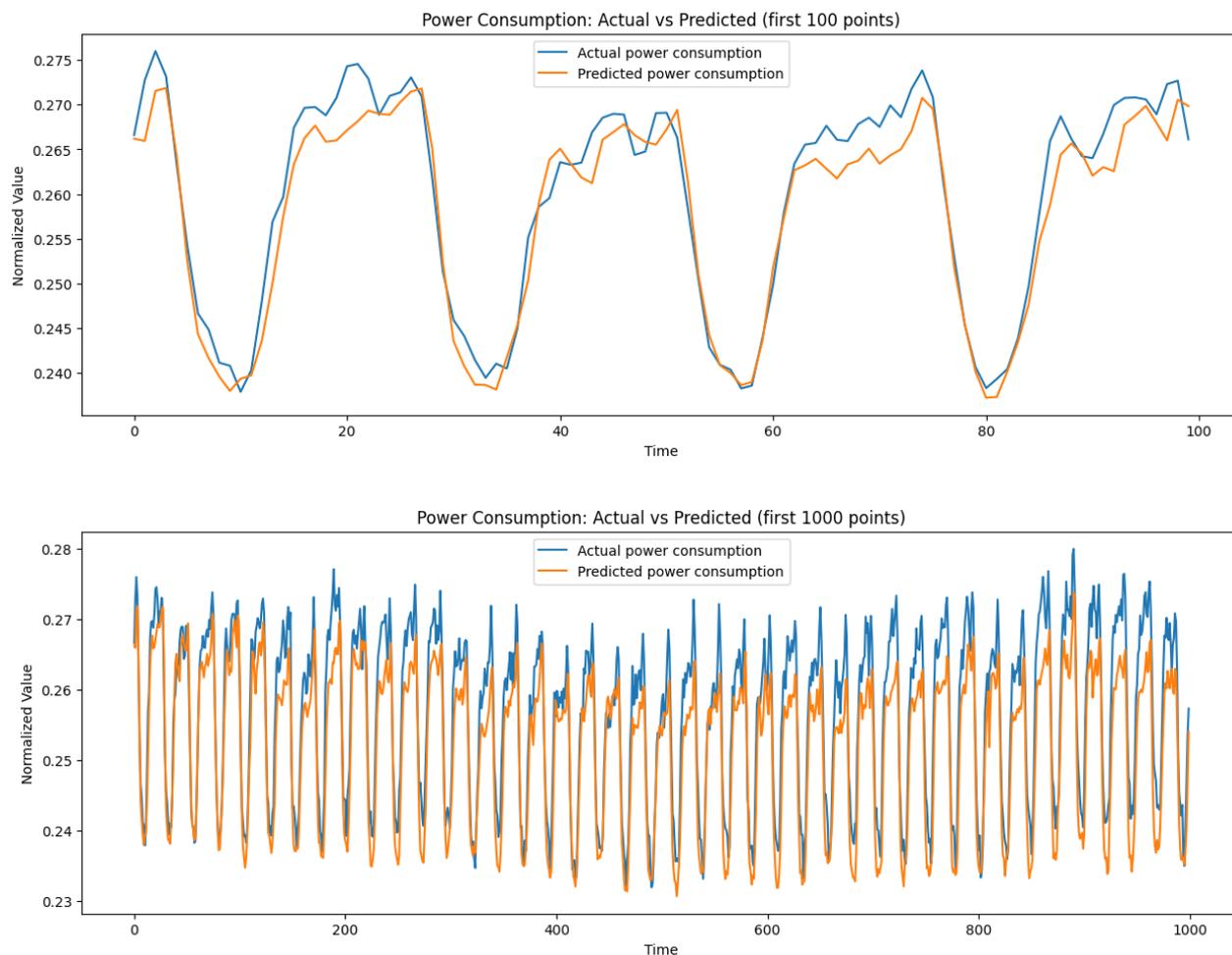


Figure 6. Actual versus Predicted graph.

The Figure 5 illustrates the comparison of actual versus predicted power consumption. Both datasets are normalized. While the predicted (in orange) and actual (in blue) trends are generally aligned, indicating a reasonably accurate prediction model, there are moments of noticeable deviation. After, the predicted values consistently overshoot the actual readings. Overall, the prediction model demonstrates a good approximation of actual power consumption with minor discrepancies at certain intervals.

Conclusion

In conclusion, methodology employed robust techniques ranging from exploratory data analysis, feature engineering, normalization, and model training, leading to the development of an optimized DRNN model. The correlation heatmap and pair plots were instrumental in understanding the dataset's underlying patterns and relationships. These insights, in conjunction with DRNN's innate ability to handle time-series data, provided a solid foundation for model training.

The model showcased commendable performance metrics with an R-squared value of 83.6%, a testament to its accuracy. However, while the RMSE and MAE provided an encouraging picture of the model's predictive capabilities, there were instances of divergence between actual and predicted values, underscoring the need for continuous refinement.

In essence, while DRNNs represent a leap forward from traditional forecasting models, the journey towards perfecting them remains ongoing. Their computational intensity and the need for large datasets can be hurdles, but the promise they hold, as evidenced by this study, is un-

deniable. The energy sector stands at the cusp of an era where data-driven insights, powered by deep learning, can significantly mitigate risks and guide infrastructural advancements. As this study on Kazakhstan's energy consumption has shown, when appropriately harnessed, such models can be a linchpin in ensuring a resilient and efficient energy future. Future research should focus on addressing DRNN limitations and diversifying datasets to enhance generalization and accuracy further.

Acknowledgement

This research paper has been funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan (AP19579354).

References

- [1] Agrawal, R.K., Muchahary, F., & Tripathi, M.M. (2018). Long term load forecasting with hourly predictions based on long-short-term-memory networks. *2018 IEEE Texas Power and Energy Conference (TPEC)*. <https://doi:10.1109/tpec.2018.8312088>
- [2] Taheri, S., Jooshaki, M., & Moeini-Aghtaie, M. (2021). Long-term planning of integrated local energy systems using deep learning algorithms. *International Journal of Electrical Power & Energy Systems*, 129, 106855. <https://doi.org/10.1016/j.ijepes.2021.106855>
- [3] Barzola-Monteses, J., Yáñez-Pazmiño, W., Flores-Morán, E., & Bravo, F. P. (2022). Comparisons of Deep Learning Models to predict Energy Consumption of an Educational Building. *2022 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT)*. <https://doi.org/10.1109/icmlant56191.2022.9996517>
- [4] Jinjin, R., Wang, S., Wang, Z., & Mei, Y. (2019). Residential Energy Use Prediction across different Time Scales with Advanced Machine Learning Techniques. *2nd Asia Conference on Energy and Environment Engineering (ACEEE)*. <https://doi.org/10.1109/aceee.2019.8816955>
- [5] Nguyen, V.H., Ga, B.V., Kim, J., & Jang, Y.M. (2020). Power Demand Forecasting Using Long Short-Term Memory Neural Network based Smart Grid. *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*. <https://doi.org/10.1109/icaaic48513.2020.9065270>
- [6] El-Taraboulsi, J., Cabrera, C., Roney, C., & Aung, N. (2023). Deep neural network architectures for cardiac image segmentation. *Artificial Intelligence in the Life Sciences*, 4, 100083. <https://doi.org/10.1016/j.aillsci.2023.100083>
- [7] Qiao, L., Li, Z., Xiao, B., Shu, Y., Wang, L., Shi, Y., Li, W., & Gao, X. (2023). QDRJL: Quaternion dynamic representation with joint learning neural network for heart sound signal abnormal detection. *Neurocomputing*, 562, 126889. <https://doi.org/10.1016/j.neucom.2023.126889>
- [8] Gomes, R., Pham, T.D., He, N., Kamrowski, C., & Wildenberg, J. C. (2023). Analysis of Swin-UNet vision transformer for Inferior Vena Cava filter segmentation from CT scans. *Artificial Intelligence in the Life Sciences*, 4, 100084. <https://doi.org/10.1016/j.aillsci.2023.100084>
- [9] Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of Recurrent Neural networks: LSTM cells and network architectures. *Neural Computation*, 31(7), 1235–1270. https://doi.org/10.1162/neco_a_01199
- [10] Paparusso, L., Melzi, S., & Braghin, F. (2023). Real-time forecasting of driver-vehicle dynamics on 3D roads: A deep-learning framework leveraging Bayesian optimisation. *Transportation Research Part C: Emerging Technologies*, 156, 104329. <https://doi.org/10.1016/j.trc.2023.104329>
- [11] Yang, X., Yu, S., & Zhou, Y. (2020). LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example. *2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)*, pp. 98-101. <https://doi.org/10.1109/IWECAI50956.2020.00027>
- [12] Ghadekar, P., Malwatkar, N., Sontakke N., & Soni, N. (2023). Comparative Analysis of LSTM, GRU and Transformer Models for German to English Language Translation. *2023 3rd Asian Conference on Innovation in Technology (ASIANCON), Ravet IN*, pp. 1-7. <https://doi.org/10.1109/ASIANCON58793.2023.10270018>

- [13] Wensheng, L., Kuihua, W., Liang, F., Hao, L., Yanshuo, W., & Can, C. (2020). A Region-Level Integrated Energy Load Forecasting Method Based on CNN-LSTM Model with User Energy Label Differentiation. *2020 5th International Conference on Power and Renewable Energy (ICPRE)*, pp. 154-159. <https://doi.org/10.1109/ICPRE51194.2020.9233226>
- [14] Abraham, A.V., Sasidharan, P., Tejas, S.J.S., Manohara, M., Muthu, R., & Naidu, R.C. (2022). Predicting Energy Consumption Using LSTM and CNN Deep Learning Algorithm. *2022 7th International Conference on Environment Friendly Energies and Applications (EFEA), Bagatelle Moka MU*, pp. 1-6. <https://doi.org/10.1109/EFEA56675.2022.10063818>
- [15] Al-Alami, H., & Jamleh, H.O. (2023). Use of Convolutional Neural Networks and Long Short-Term Memory for Accurate Residential Energy Prediction. *2023 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, pp. 294-299. <https://doi.org/10.1109/JEEIT58638.2023.10185888>
- [16] Hamayat, F., Akram, Z., & Zubair, S. (2023). Deep Learning-based Predictive Modeling of Building Energy Usage. *2023 6th International Conference on Energy Conservation and Efficiency (ICECE)*, pp. 1-6. <https://doi.org/10.1109/ICECE58062.2023.10092502>
- [17] Alhussein, M., Aurangzeb, K., & Haider, S.I. (2020). Hybrid CNN-LSTM Model for Short-Term Individual Household Load Forecasting. *IEEE Access*, 8, pp. 180544-180557. <https://doi.org/10.1109/ACCESS.2020.3028281>
- [18] Shen, H., Wang, Z., Zhou, X., Lamantia, M., Yang, K., Chen, P., & Wang, J. (2022). Electric Vehicle Velocity and Energy Consumption Predictions Using Transformer and Markov-Chain Monte Carlo. *IEEE Transactions on Transportation Electrification*, vol. 8, no. 3, pp. 3836-3847. <https://doi.org/10.1109/TTE.2022.3157652>
- [19] Hu, P., Shan, X., Hu, D., Fu, J., Wang, C., & Zhang, K. (2022). Source-Load Joint Probability Prediction Based on Transformer Model. *2022 5th International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pp. 235-240. <https://doi.org/10.1109/ICAIBD55127.2022.9820171>
- [20] Yuan, J., Chen, S.-Z., Yu, S.S., Zhang, G., Chen, Z., & Zhang, Y. (2023). A Kernel-Based Real-Time Adaptive Dynamic Programming Method for Economic Household Energy Systems. *IEEE Transactions on Industrial Informatics*, vol. 19, no. 3, pp. 2374-2384. <https://doi.org/10.1109/TII.2022.3181034>