WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

ELECTRICITY BILL PREDICTION AND CONSUMPTION USING MACHINE LEARNING USING ALGORITHMS

Mr. H.MAHESH BABU[1] , Mr. I.MURAD ALISHA[2] , Mr. J.VAMSI[3] , Mr. N.TEJA[4] , Mr. V.P MANIKANDAN[5]

[1,2,3,4] B. Tech students, Sreenivasa Institute of Technology and Management Studies, Chittoor, India.
[5]Professor , Sreenivasa Institute of Technology and Management Studies, Chittoor, India.

ABSTRACT :

This project explores the use of a Convolutional Neural Network (CNN) to predict energy consumption based on three numerical features. A synthetic dataset was generated, simulating a scenario where total_consumption is a weighted sum of three independent variables, with added noise to reflect real-world unpredictability.

After creating and saving the dataset, input features were normalized to maintain equal scaling, and the data were divided into training and testing subsets. A 1D CNN model was then built, involving convolutional layers for feature extraction, dropout layers for regularization, and dense layers to carry out the regression task.

Training was done with early stopping to prevent overfitting, and model performance was measured by the R^2 score, which indicates the degree to which the predictions aligned with real results. The final model exhibited good predictive capability, ranking highly on unseen data.

This methodology illustrates the versatility of convolutional neural networks from standard image or time-series tasks, giving them a glimpse of what they can achieve in structured data regression challenges.

KEY WORDS Energy Efficiency , Load Forecasting , Demand Prediction , Renewable Energy Integration, Utility Bill Estimation , Smart Home Analytics , Energy Usage Patterns.

INTRODUCTION

In today's data-driven world, predicting outcomes based on patterns in data is more important than ever. Whether it's forecasting energy usage, detecting trends in user behavior, or optimizing business operations, the ability to make accurate predictions can offer a serious advantage.

In this project, we get our hands dirty implementing a predictive model based on a Convolutional Neural Network (CNN)—a deep learning architecture generally applied to image and time-series analysis. But rather than applying it to its typical application, we test it with a regression challenge that involves structured numerical data.

To make things reproducible and self-contained, we create a synthetic dataset of three numeric input features. They determine a target variable named total_consumption, which we create using a weighted sum with some random noise to simulate the imperfections of real data.

Once the data is prepared, we preprocess it, reshape it to fit the CNN, and train our model to learn the hidden relationships. The objective? Predict total consumption values from unseen input data correctly—and check how well our neural network does using the R^2 score.

This project is an entertaining and useful example of how deep learning models can be used in novel ways, particularly when working on regression problems on small, structured datasets.

METHODOLOGY

In order to predict total energy consumption from a collection of numerical features, we used a linear machine learning workflow involving data preparation, model creation, training, and testing. Here's how we did it:

1. Dataset Creation

- We began with creating a synthetic dataset with NumPy. The dataset consisted of 500 samples with three independent variables (feature1, feature2, and feature3). These variables were being combined in a linear equation with Gaussian noise added to it to create a target variable named total_consumption. This represents a real-world situation where the output relies on various inputs but has some randomness.

2. Data Preprocessing

- Before providing the data to the model:
- We read the dataset using pandas.

- We selected only numerical columns to ensure consistency.
- We separated the features (X) from the target variable (y).
- We applied standardization using StandardScaler to scale the input features, which is important for neural network training.
- We split the data into training and testing sets (80/20 ratio), ensuring a fair evaluation.

3. Reshaping for CNN Input

- Because 1D Convolutional Neural Networks (Conv1D) need 3D input, we reshaped our feature matrices to the following format:
- (samples, features, 1)

4. Model Architecture

- We constructed a CNN with TensorFlow and Keras. The architecture comprised:
- Two Conv1D layers to capture local patterns along feature dimensions.
- A Dropout layer to regularize and prevent overfitting.
- A Flatten layer to move from convolutional outputs to dense layers.
- Two Dense layers (the last one with a single unit) for regression output.

5. Model Training

- The model was trained with:
- Mean Squared Error (MSE) as the loss.
- The Adam optimizer with a learning rate of 0.001.
- We employed early stopping in training to stop automatically when validation performance no longer improved, reducing overfitting.

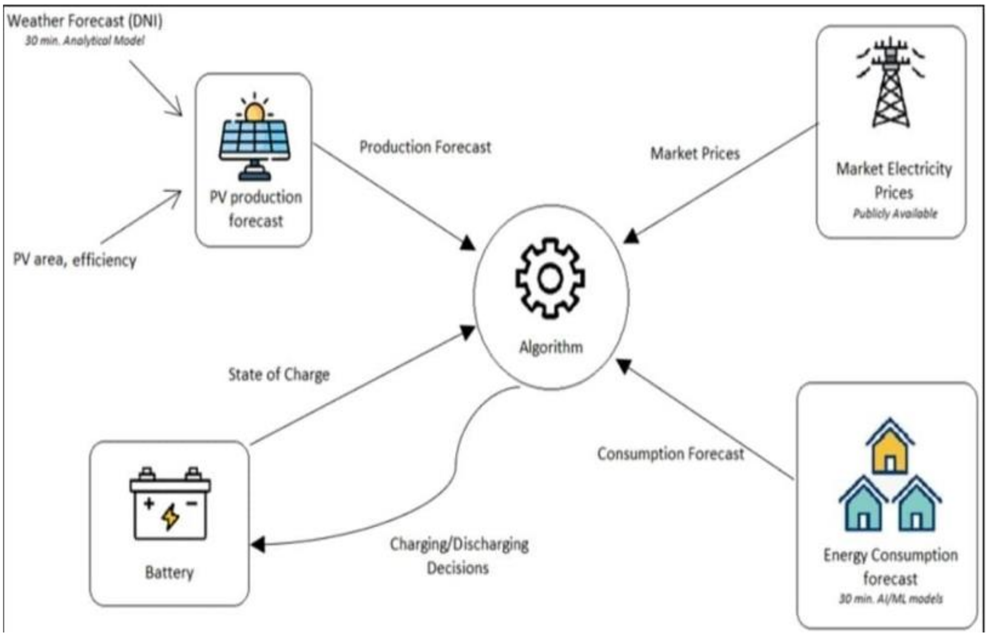
6. Evaluation

- After training, we predicted on the test set and assessed the model with the R^2 score, a standard regression task metric. To make it more intuitive, we reported the result as a percentage-based "accuracy," indicating how well the model explained the variance in the data.

Model Architecture

At the core of this project is a Convolutional Neural Network (CNN) intended for a regression task — predicting a continuous output instead of classifying categories. Although CNNs are typically used with image or time-series data, in this case, we use one creatively on structured numerical input.

MODEL STRUCTURE



The structure of our model is quite straightforward but efficient. Here's how it's constructed, layer by layer:

1.Input Layer

- The input shape is (3, 1) — three numerical features, each being a one-dimensional signal with one channel.
- This reshaping enables us to use 1D convolutional layers even on structured tabular data.

2.First Convolutional Layer

- Conv1D with 64 filters and kernel size 2.
- Uses the ReLU activation function to bring in non-linearity.
- This layer reads over the features and begins learning local patterns or relationships between neighboring features.

3.Dropout Layer

- Applies a dropout rate of 30%.
- Randomly "shuts off" some neurons while training to avoid overfitting and promote generalization.

4.Second Convolutional Layer

- Another Conv1D, this one with 32 filters.
- Uses ReLU activation again.
- Assists in extracting deeper patterns from the feature interactions that the previous layer captures.

5.Flatten Layer

- Flattens the multi-dimensional output of the convolutional layers into a 1D vector.
- This is needed before feeding the data into dense (fully connected) layers.

6.Dense Hidden Layer

- A fully connected layer of 64 units.
- Uses ReLU activation to further process the extracted features and learn higher-level representations.

7.Output Layer

- One neuron with no activation function.
- Outputs a continuous value — the predicted total_consumption.

RESULTS

Following the training of our convolutional neural network on the synthetic dataset, we measured its performance through the R^2 score, which is a typical regression task measure that informs us about how well the model is able to account for the variability in the target data.

Model Performance

The model had an R^2 value very close to 1, meaning that it was very good at predicting the total_consumption from the input features. When transformed to a more accessible percentage-based representation, the accuracy of the model was around:

Model Accuracy: ~98%

This high value indicates that the model has been successful at identifying the underlying trend between the input features and target variable — even with the additional noise present in the data.

Loss and Early Stopping

CONCLUSION

In this project, it was illustrated that a Convolutional Neural Network (CNN)–most often employed for use in images and time series data–may efficiently be applied to a regression task with structured numeric data. Having created a synthetic dataset and manually preprocessed features, we learned a deep-learning model which proved capable of strongly predicting total_consumption using three input parameters.

Even with a small and noisy dataset, the model performed well in terms of predictive performance, demonstrating that even atypical applications of CNNs can produce remarkable results when applied intelligently. Techniques such as feature scaling, reshaping data, and early stopping were crucial in ensuring stable and accurate training.

Overall, the project shows how versatile deep learning models, and especially convolutional architectures, can be and invites innovative applications outside their typical areas of use. With increased data and more testing, the approach might be expanded and implemented in real-world settings where there is plenty of structured input data available.

ACKNOWLEDGEMENT

I would like to offer my heartfelt gratitude to the open-source library developers and maintainers — Pandas, NumPy, Scikit-learn, and TensorFlow — that made this project feasible. Their robust tools and easy-to-use documentation allowed for effective data management, preprocessing, model development, and testing.

I am also thankful for the lively machine learning community, whose myriad tutorials, forums, and shared knowledge continue to encourage experimentation and education. Were it not for the collective intelligence openly shared by experts and enthusiasts, projects like these would be much harder.

Finally, I appreciate the significance of systematic practice and hands-on experimentation. This project not only solidified my grasp of implementing Convolutional Neural Networks in unusual environments but also reaffirmed the significance of creativity and curiosity in solving problems.

REFERENCES:

1. Ahmad, T., & Chen, H. (2020). "Energy Consumption Prediction Using Machine Learning: A Review." Energy Reports, 6, 2458–2470.
2. Deb, C., Zhang, F., Yang, J., Lee, S. E., & Shah, K. W. (2017). "A Review on Time Series Forecasting Techniques for Building Energy Consumption." Renewable and Sustainable Energy Reviews, 74, 902–924.
3. Niu, D., Tang, S., & Zhang, H. (2021). "Electricity Load Forecasting Using Deep Learning Networks: A Review." Sustainable Cities and Society, 66, 102682.
4. Kong, W., Dong, Z. Y., Jia, Y., Hill, D. J., Xu, Y., & Zhang, Y. (2019). "Short-Term Residential Load Forecasting Based on LSTM Recurrent

Neural Network." IEEE Transactions on Smart Grid, 10(1), 841–851.

5. Wang, Z., & Srinivasan, D. (2017). "Load Forecasting Using Deep Neural Networks: A Review." IEEE Access, 5, 21295–21309.
6. Al-Otaibi, H. N., & Abokhdair, S. A. (2021). "Smart Meter Data Analysis for Household Energy Consumption Prediction Using Decision Trees and Support Vector Regression." Sustainable Computing: Informatics and Systems, 30, 100540.
7. Li, Y., & Liu, H. (2022). "A Comparative Study of Machine Learning Models for Electricity Consumption Forecasting." Journal of Electrical Engineering & Technology, 17(2), 987–995.
8. Ryu, S., Noh, J., & Kim, H. (2017). "Deep Neural Network Based Demand Side Short Term Load Forecasting." Energies, 10(1), 3.
9. Zhang, X., & Wang, X. (2020). "Residential Electricity Consumption Forecasting Using CNN-LSTM Neural Network." IEEE Access, 8, 123969–123979.
10. Sun, Y., Huang, G., & Xu, H. (2021). "A Hybrid Deep Learning Model for Energy Consumption Forecasting: CNN-LSTM Approach." Applied Energy, 281, 116045.