IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 8, Issue 03, October 2019)

Load Prediction using Deep Reinforced Learning

Prashant Dutta
Programmer, MPPKVVCL, Jabalpur
prashantdutta786@gmail.com

***Abstract*— The world is moving towards energy conservation every new day. However to conserve energy we should first manage and predict energy consumption. The 'Artificial Intelligence' world is constantly making its efforts to develop accurate load forecasting algorithms. Machine-learning algo's can predict for load with great precision. Even companies like Amazon-AWS and Google-Azure has its own AI and ML tools to compliment load forecasting using its own cloud based resources. This paper proposes an approach to predict power consumption in a colony. The proposed method consists of five different layers, namely data retrieval, Data Cleaning, Data Filtering, Machine Learning & Forecasting and Assessment. We have used the deep reinforcement learning for predicting the energy consumption. In the assessment layer K-cross validation is used for validating the Deep Reinforcement learning. The prediction is validated using the 5 dissimilar trials namely: Misclassification Rate, Correctness, Completeness, and Effectiveness & Efficiency.**

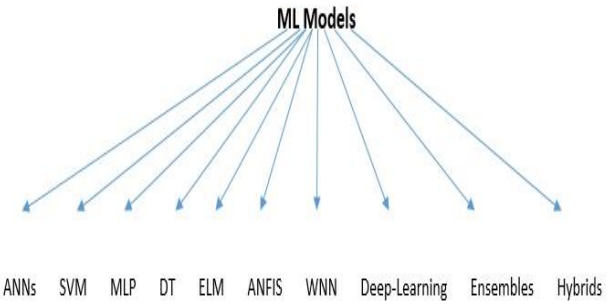
***Keywords*— Machine learning, Power consumption, load forecast, Deep Learning, A.I.**

I. INTRODUCTION

Load Prediction for residential-colonies can help a lot for power management and in turn can form a bridge between Power generating companies and power distribution companies. Across the world the Power sector has mainly 3 types of sub-companies namely Generation, Transmission and Distribution. The power generation company produces electricity through hydel, coal, nuclear, wind etc. Then the power transmission company transmits that power from generating end to the sub-station of the big cities. From the sub-station the distribution companies take the charge and distribute the electricity to the consumers. The Power generating companies need a feedback from the distribution companies about how much power has to be generated in the coming months. This prediction is normally done by the distribution company on the basis of its previous history and is mostly inaccurate. Hence there comes the need for accurate load prediction. This paper showcases a 5 –tier-model for forecasting load demand of colony of 100 homes using Deep reinforcement learning (DRL) .Machine-learning has underwritten many futuristic forecasting-models for load prediction.

A. Machine Learning

Machine learning is a branch of 'AI' that delivers computer-software's the capacity to automatically learn and improve from past-experience and they need not be programmed for that. Basically machine learning wants the software program to develop its own brains through adaptive learning. To achieve this scientists have started training the software programs just like human minds. Hence the machine learning paradigm consist of neural networks just like neurons present in human brains. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.



B. Some Machine Learning Methods

Machine-learning algos have been classified supervized or unsupervized.

Supervised ML algos can to what has been educated in the past to fresh-data using instances to predict future actions. By analyzing the training dataset, the ML Algo generates a contingent function to create predictions about the output-values.

However unsupervised ML algos are taken into account where the info used to train the algo is not categorized. The model doesn't identify the accurate output, but it discovers the information and can comprehend from data-sets to label concealed arrangements from un-labeled data.

IJDACR
ISSN: 2319-4863

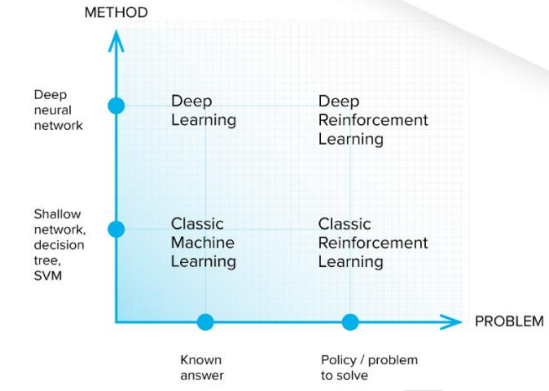
IJDACR
International Journal Of Digital Application & Contemporary Research

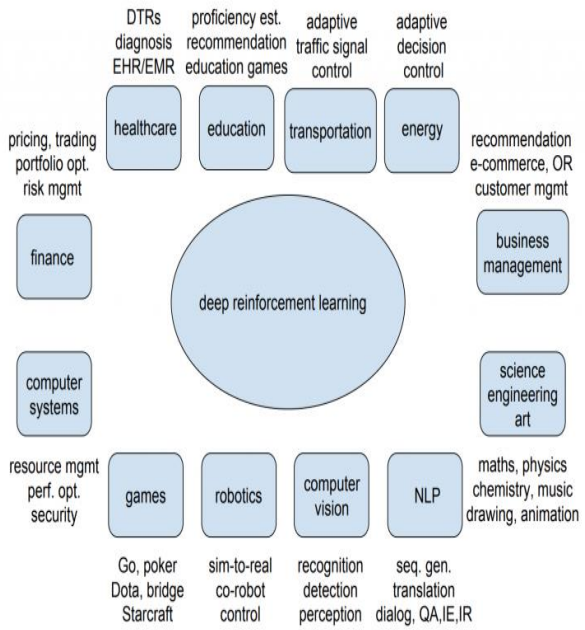
International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 8, Issue 03, October 2019)

Reinforced ML algos is a learning method that communicate with its surroundings by creating actions and learns mistakes or rewards. Trial-&-error search and late reward are the best appropriate physiognomies of reinforced learning.

1.3 Deep reinforcement learning

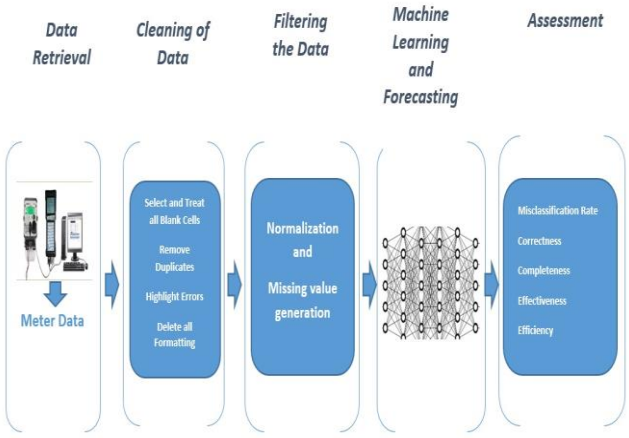
Deep-reinforcement-learning('DRL') employs deep-learning and reinforced learning doctrines to generate effective algos that can be implemented on fields like robotics, video-games, load predictions and health-care. Applying deep-learning design (deep-neural-networks) with reinforcement-learning-algorithms, a strong model ('DRL') can be generated that is proficient to gauge to formerly insoluble problems. DRL employs raw-sensor or image-signals as input and can receive the advantage of end to end reinforcement-learning as well as that of convolutional-neural-networks.





II. THE PREDICTION MODEL

This paper proposes a 5 step process in which the second last step is the forecasting step and the last and final step is the assessment step to evaluate the predictions done in the second last step. The forecasting is done through Deep reinforcement learning (DRL).



A. Data Retrieval Step

The billing data was collected from a colony of 100 residents for the period from Jan to Dec 2019. The bill data was obtained using the IVRS. The IVRS serves as a unique identifier of the consumer. This billing data serves as the input for the Data Cleaning Step.

B. Data Cleaning Step

Data-cleansing or data-cleaning is the procedure of spotting and altering (or removing) shady or inexact data from a record-set, table, or data-base and states to finding imperfect, incorrect, imprecise or immaterial parts of the data and then substituting, altering, or erasing the unclean or rough data. Data-cleansing may be achieved with data squabbling tools, or as batch-processing via scripting. In our case, the monthly bills are sometimes generated on Average-Reading Basis when the consumer premises is not accessible for data reading. Hence these Average–reading data needs to be cleansed.

C. Filtering the Data

In our case we have considered only Domestic meters hence any commercial or agricultural meter data are filtered out.

D. Load Prediction

1) RL Agent-Environment

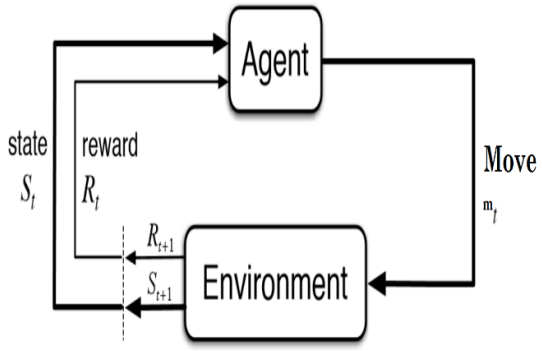
A reinforced learning job is about training an agent which intermingles with its surroundings. The agent reaches at

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 8, Issue 03, October 2019)

diverse situations known as 'states' by execution of 'moves'. Moves clue to 'rewards' which can be good and bad.
The 'agent' has only one motive– to maximize its total 'reward' through an 'episode'. This 'episode' is anything and everything that happens between the first state and the last or terminal state within the environment. We reinforce the agent to learn to perform the best moves by experience. This is the 'strategy or policy'.



2) R Learning
Assume that the predictable reward of every 'Move' at each phase. This will be as a guessing-paper for the 'agent'. The 'agent' will discern which 'move' to execute.
The agent will now perform only those sequence of 'moves' that will finally produce maximum 'reward'. This total-reward is also known as the R-value and can be coined as:

$$R(s, m) = r(s, m) + \xi \max_m R(s', a)$$

The equation stated as above says the R-value can be generated from state 's' and by executing move 'm' with instant reward r(s,m) in addition the uppermost R-value likely from the next state s'. Xi(ξ) is the 'discount-factor' which panels the involvement of rewards in the near future. R(s',a) again depends on R(s",a) which will then have a coefficient of Xi(ξ) squared. So, the R-value depends on R-values of future-states as shown here:

$$R(s, m) \rightarrow \xi R(s', m) + \xi 2R(s'', m) \text{ -- -- --}$$
$$\text{----------}\xi nR(s''—n, m)$$

Iterating the value of Xi(ξ) will lessen or amplify the contri of future 'rewards'.
As this is a recursive-equation, we need to initiate by taking random predictions for all 'r-values'. With time, it will meet to the ideal strategy. In real-world states, this is applied as :

$$R(s_t, m_t) \leftarrow R(s_t, mt) + \eta [\check{R}t_{+1} + \xi \max_m Q(s_{t+1}, m) - R(s_t, m_t)]$$

where Ř is the 'learning-rate' or 'step-size'. This merely governs to what degree newly learned data overrides old data.
'R-learning' is a modest hitherto fairly potent algo to make a 'cheat-sheet' for our 'agent'. This aids the 'agent' to envisage which event to execute. However problem arises when the cheat-sheet is large. Assume a paradigm with 20,000 states and 1500 actions per-state. This would make a table of N million booths. It can be clearly comprehended that it is next to impossible to infer the 'R-value' of new-states from already discovered states. This showcases 2 difficulties:
-> The memory needed to handle so many states increases.

-> The total time needed to discover each state to make the required Q-table would be impractical.

3) Deep R-Networks

The deep-R learning uses neural-network to extrapolate the 'R-value' function. Input is provided in the form of state and output is generated in the form of 'R-value'.
Phases embedded in reinforcement-learning using 'deep R-learning networks' (DRNs):

- The memory is used as a storage for past user-experiences.
- The output from the 'R-network' will decide the next course of action.
- The loss-function is mean-squared-error of the foretold 'R-value' and the target 'R-value' – R*. This convolutes to a Regression Paradigm. But the 'Target/Actual' value is not known as we are interacting with a reinforced-learning-problem.

The updated equation is now as under:

$$R(s_t, m_t) \leftarrow R(s_t, mt) + \eta [\check{R}_{t+1} + \xi \max_m Q(s_{t+1}, m) - R(s_t, m_t)]$$

The area in red represents the target. We can debate that it is forecasting its own value, but since R is the unbiased-true-reward, the network is going to update its slope using 'backpropagation' to finally convolute.

E. Assessment
Cross-validation is a method to evaluate predictive-models by splitting the original-sample into a training-set to train the model, and a test set to evaluate it. In k-fold cross-validation, the original sample is arbitrarily segregated into 'k' same size subsamples. Of the 'k' sub-samples, a single sub-sample

IJDACR
ISSN: 2319-4863

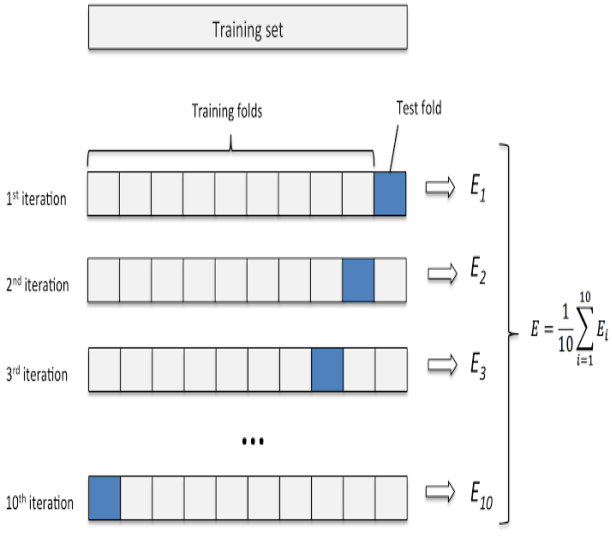
IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 8, Issue 03, October 2019)

is reserved as the validation-data for testing the model, and the remaining k-1 subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k sub-samples used exactly once as the 'validation-data'. The k results from the folds can then be averaged to create a single approximation. The benefit of this technique is that all observations are used for both 'training and validation', and each statement is used for validation exactly once. For sorting problems, one typically uses stratified k-fold cross-validation, in which the folds are nominated so that each fold contains coarsely the same proportions of class-labels.

In repeated-cross-validation, the cross-validation procedure is repeated 'n' times, yielding 'n' random partitions of the original sample. The 'n' outputs are again averaged to generate a single estimation.

"Open-ML" creates train-test splits given the number of folds and reiterates, so that different users can evaluate their models with the same splits. Stratification is applied by default for classification problems (unless otherwise specified). The splits are given as part of the task description as an ARFF file with the row id, fold number, repeat number and the class (TRAIN or TEST). The uploaded predictions should be labeled with the fold and repeat number of the test instance, so that the results can be properly evaluated and aggregated. OpenML stores both the per fold/repeat results and the aggregated scores

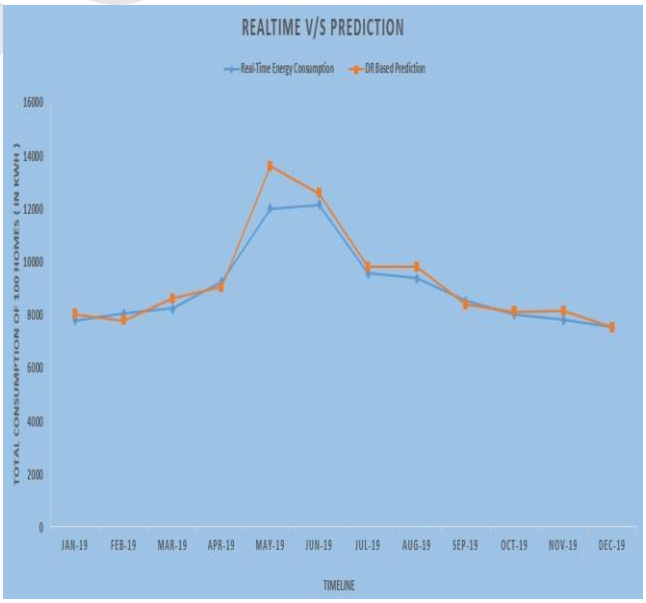


III. Model Validation of Deep Learning

To evaluate the DR Model we have gathered realtime data from a colony of 100 residents for the period from Jan to Dec 2019. The bill data was obtained using the IVRS. The IVRS serves as a unique identifier of the consumer. The units consumed of each month of each consumer is recorded in kwh. The ML algo is trained using the meter data of each consumer for a particular day. Further we have used K-fold validation because it guarantees that every record from the original data-set has the opportunity to appear in training and test-set.

To predict the load for one month the data-set has been split into twelve-k sets of equal size. To start with we have used the January month data for testing and the remaining eleven months for training. Similarly in the next iteration we have used February month data for testing and rest eleven months for training. In the same pattern, for the 12th iteration we have used the December month data for testing and the rest eleven month's data for training. At last the average was conceived from all the data. To obtain best structure we perceived the trial and error method. We landed up to find the 7 inner layers with 21 neuron in each inner layer for the projected Deep Reinforcement learning method.

We have used the Gaussian function as our activation function as it is widely used by various programmers for deep learning. Using the optimal model we have plotted the monthly load consumption between actual and deep learning prediction as shown below:



IV. Conclusions and Future Work

Previously Load forecasting was done using very orthodox methods like "Trend Analysis" & "End-User Analysis". However after the advent of Artificial Intelligence and Machine Learning Paradigms load forecasting has seen an altogether different dimension. In this paper we have

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 8, Issue 03, October 2019)

proposed an advanced version of AI using Deep Reinforcement learning to forecast future load.

However we have taken into account only the meter data but going ahead in the future we need to take into account several other factors like Meter make and model, climatic conditions and government policies. These factors will help us to reach to a more precise prediction. Also going ahead we would like to use the AWS and Azure cloud platforms as our workbench which shall help us to work in a more faster and efficient way.

REFERENCES

[1] K. Singh, M. Bhadauria, S.A. McKee, Real time power estimation and thread scheduling via performance counters, ACM SIGARCH Comput. Archit. News 37 (2) (2009) 46–55.

[2] C. Spearman, The proof and measurement of association between two things, Am. J. Psychol. 15 (1) (1904) 72–101.

[3] V. Spiliopoulos, A. Sembrant, S. Kaxiras, Power-sleuth: A tool for investigating your program's power behavior, in: Modeling, Analysis & Simulation of Computer and Telecommunication Systems, MASCOTS, 2012 IEEE 20th International Symposium on, IEEE, 2012, pp. 241–250.

[4] Streimikiene, D. Residential Energy Consumption Trends, Main Drivers and Policies in Lithuania. Renew. Sustain. Energy Rev. 2014, 35, 285–293.

[5] Zuo, J.; Zhao, Z.Y. Green Building Research-Current Status and Future Agenda: A review. Renew. Sustain. Energy Rev. 2014, 30, 271–281.

[6] Prashar, A. Adopting PDCA (Plan-Do-Check-Act) Cycle for Energy Optimization in Energy-Intensive SMEs. J. Clean. Prod. 2017, 145, 277–293.

[7] Collobert, R.; Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 160–167.

[8] Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. Science 2006, 313, 504–507. [CrossRef] [PubMed]

[9] Nau, R. Forecasting with Moving Averages. Fuqua School of Business, Duke University, 2014. Available online: https://people.duke.edu/~rnau/Notes_on_forecasting_with_moving_averages--Robert_Nau.pdf .

[10] Ciarelli, Patrick Marques, Oliveira, Elias, 2009. An enhanced probabilistic neural network approach applied to text classification. In: LNCS, 5856. Springer Verlag, pp. 661–668.

[11] Dave, K., Lawrence, S., Pennock, D.M., 2003. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In: The 12th WWW.

[12] Gamon, M., 2004. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In: Proceedings of the 20th International Conference on Computational Linguistics, p. 841.