Deep Learning Approach cum Aggregated Smart Meter Data Based Residential Energy Load Modeling

J. SHANMUGASUNDARAM[1], K. INDHUMATHI[2], S. SIVARANJANI[3]
[1]*Assistant, Vivekanandha College of Engineering for Women, Tiruchengode*
[2, 3] *PG Student, Vivekanandha College of Engineering for Women, Tiruchengode*

Abstract- Conventional load forecasting dedicated to the quantity of load growth. The target of spatial load forecasting would be to estimate with reasonable accuracy, and a advanced level of geographic resolution, not just the quantity of load growth, but additionally when and where new load will effective locations for distribution facilities and to plan system growth (e.g., new substations, distribution feeders, transformers, and so on). Spatial load forecasting is conducted on the foundation of small areas, historical load and weather data, land use, and geographic information. The proposed forecasting methods are classified into two major categories: (i) Univariate (time series) forecasting models and (ii) Multivariate forecasting models. The forecasting data classified and analyzed by utilizing Support Vector Machine algorithm. Comparative analysis of those methods can be done in this survey. Furthermore, the forecasting techniques are reviewed from the areas of big data and conventional data by utilizing deep learning approach.

Indexed Terms- SVM (Support Vector Machine), Spatial Load Forecasting and Deep Learning.

I. INTRODUCTION

The uses of extremely large data sets in power system operation, control, and protection, which are difficult to process with traditional database tools in many cases are termed as big data. This really is an emerging technical problem brought with a dataset of large volume, various categories and complicated structures which needs novel framework and techniques to excavate useful information effectively. Therefore, this is of big data depends upon the capability of data mining algorithms and the corresponding hardware equipment to cope with large volume datasets. Smart grid is the energy system embedded by having an information layer which allows for two-way communication involving the central controllers and local actuators along with logistic units to respond digitally to urgent situations of physical elements or quickly changing of electric demand. In this paper to develop and implement a simulation model for a Smart meter Data Management for Residential Load data using Support Vector Machine (SVM) based deep learning technique.

This algorithm manages the huge amount of metering data and clusters the data for efficient load and meter forecasting with the cost operation by utilizing the time of use electricity pricing concept. The primary objectives of this project is to predict or forecast the future cost estimation or the load utilization for the same building or group of buildings based on the historical data stored and managed on the weekly basis and provide to the consumers for efficient utilization. This algorithm has been implemented to improve the efficiency of Big data management with the smart power grid management system.

II. PROPOSED SYSTEM FUNCTION

The system consists of Power source, smart meter, Data Base management system and Load. The electrical power consumed by the power source by each individual building or industry connected from the main grid is measured using the smart energy meters. The readings of power consumption are made on the smart meter units is based on the number of loads connected and the time for which the load consumes power. These data of smart meter are managed in the data base or data store for further processing. The data collected from the meters are if huge amount of data that has to be managed and processed efficiently. The large amount of data can be classified and processed using a classifier with predictive machine learning algorithm called Support

Vector Machine. This process is shown in the below Fig.2.

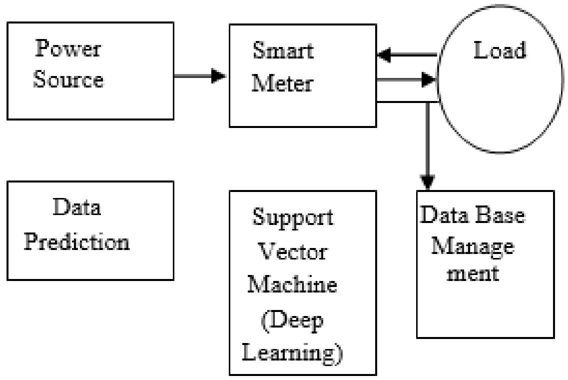

Fig.2. Functional block diagram of system

Machine learning is a multi-disciplinary field, involving many disciplines such as for example probability theory, statistics, approximation theory, convex analysis, and algorithm complexity theory. Specializing in how computers simulate or realize human learning behavior to obtain new knowledge or skills, and reorganize existing knowledge structures to continuously boost their performance. The device learning is put on the calculation and classification of data acquired from the smart meters. Repeated learning through the device improves the accuracy of calculations. Based on the learning set outcomes of the information processed the algorithm provides the prediction data for the smart metering system.

III. ALGORITHM

1) Deep Learning

Deep learning is a class of machine learning algorithms that uses multiple layers to progressively extract higher level features from the raw input. Most contemporary deep learning models derive from artificial neural networks, specifically, Convolutional Neural Networks (CNN) s, although they are able to also include propositional formulas or latent variables organized layer-wise in deep generative models including the nodes in deep belief networks and deep Boltzmann machines.

In deep learning, each level learns to transform its input data into a slightly more abstract and composite representation. In an image recognition application, the raw input might be a matrix of pixels; the initial representational layer may abstract the pixels and encode edges; the second layer may compose and encode arrangements of edges; the third layer may encode a nose and eyes; and the fourth layer may notice that the image has a face. Importantly, a heavy learning process can learn which features to optimally place in which level on its own. (Of course, this does not completely eliminate the necessity for hand-tuning; as an example, varying numbers of layers and layer sizes can provide different examples of abstraction).

The term "deep" in "deep learning" identifies the amount of layers through that the data is transformed. More precisely, deep learning systems have a considerable credit assignment path (CAP) depth. The CAP may be the chain of transformations from input to output. CAPs describe potentially causal connections between input and output. For a feed forward neural network, the depth of the CAPs is that of the network and is the amount of hidden layers and one (as the output layer can be parameterized). For recurrent neural networks, in which a sign may propagate via a layer over and over again, the CAP depth is potentially unlimited. No universally agreed upon threshold of depth divides shallow learning from deep learning, but most researchers agree that deep learning involves CAP depth more than 2. CAP of depth 2 has been shown to be always a universal approximate in the sense that it can emulate any function.[citation needed] Beyond that, more layers don't increase the function approximate ability of the network. Deep models (CAP > 2) can extract better features than shallow models and hence, extra layers aid in learning the features effectively.

Deep learning architectures can be constructed with a greedy layer-by-layer method. Deep learning really helps to disentangle these abstractions and pick out which features improve performance. For supervised learning tasks, deep learning methods eliminate feature engineering, by translating the information into compact intermediate representations comparable to principal components, and derive layered structures that remove redundancy in representation. Deep learning algorithms can be put on unsupervised learning tasks. That is an important benefit because unlabeled data are far more abundant compared to the labeled data.

2) Support Vector Machine

In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data useful for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, which makes it a non- probabilistic binary linear classifier (although methods such as for example Platt scaling exist to make use of SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the samples of the separate categories are divided by way of a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a group on the basis of the side of the gap on which they fall.

Along with performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high- dimensional feature spaces. When data are unlabelled, supervised learning is difficult, and an unsupervised learning approach is necessary, which attempts to get natural clustering of the information to groups, and then map new data to these formed groups. The support-vector clustering algorithm, produced by Hava Siegelmann and Vladimir Vapnik, applies the statistics of support vectors, developed in the support vector machines algorithm, to categorize unlabeled data, and is one of the very most popular clustering algorithms in industrial applications.

Linear support vector machines (SVM) is originally formulated for binary classification. Given training data and its corresponding labels

$$(Xn,Yn),n=1,...,N,Xn\in RD,tn\in -1,+1,$$

SVMs learning includes the next constrained optimization:

$$\text{Min } \tfrac{1}{2}\, wTw+C\sum \xi n$$

Show that

$$w^{T}Xntn>1-\xi n\forall n \;\; \xi n\geq 0\forall n$$

ξn are slack variables which penalizes data points which violate the margin requirements. Note that individuals can range from the bias by augment all data vectors Xn with a scalar value of 1. The corresponding unconstrained optimization problem is the next:

$$\text{Min}\tfrac{1}{2}w^{T}w+C\sum \max(1-w^{T}Xntn,0)$$

The aim of the above mentioned equation is recognized as the primal form problem of L1-SVM, with the typical hinge loss. Since L1-SVM is not differentiable, a favorite variation is recognized as the L2-SVM which minimizes the squared hinge loss:

$$\text{Min } \tfrac{1}{2}\, w^{T}w+C\sum \max(1-w^{T}Xntn,0)^{2}$$

L2-SVM is differentiable and imposes a bigger(quadratic vs. linear) loss for points which violate the margin. To predict the class label of an examination data x:

$$\arg\max(wTx)t$$

For Kernel SVMs, optimization should be performed in the dual. However, scalability is a trouble with Ker-nal SVMs, and in this paper we are only using linear SVMs with standard deep learning models.

Multiclass SVMs

The simplest way to extend SVMs for multiclass problems is using the so-called one-vs- rest approach. For K-class problems, K-linear SVMs will be trained independently, where in actuality the data from the other classes form the negative cases.

Denoting the output of the k-th SVM as

$$ak(x) =w^{T}x$$

The predicted class is arg (maxak(x))

Deep Learning with Support Vector Machines currently, deep learning for classification using fully connected layers and convolutional layers have almost always used soft max layer objective to learn the lower level parameters. You can find exceptions, notably the supervised embedding with nonlinear NCA, and semi-supervised deep embedding. The multiclass SVM's objective is to train deep neural nets for classification tasks. Lower layer weights are learned by back propagating the gradients from the SVM. To achieve this, we need to differentiate the SVM objective regarding the ac-tivation of the penultimate layer.

$$\partial l(w)/\partial hn=-Ctnw(I1>w^{T}hntn)$$

Where I may be the indicator function. Likewise, for the L2-SVM,

$\partial l(w)/\partial h_n = -2Ct_n w(\max(1-w^T h_n t_n, 0))$

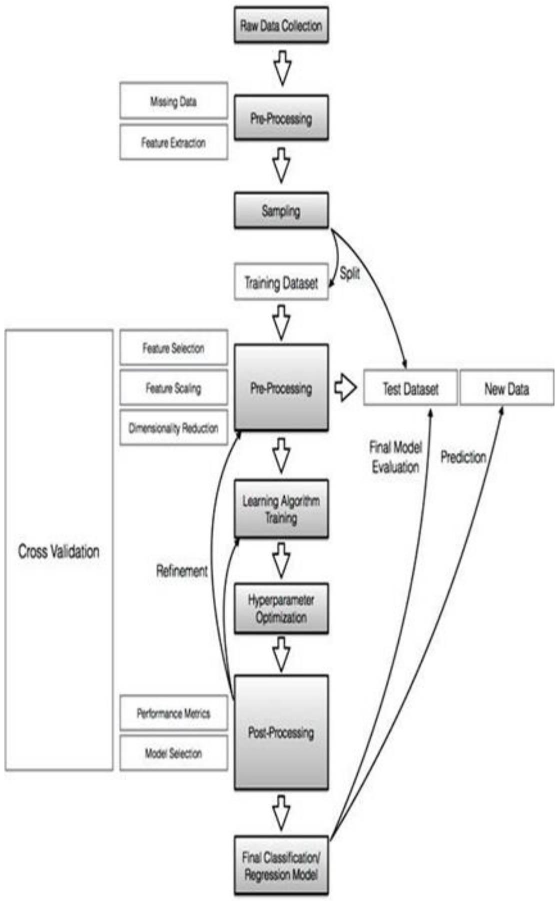

Fig.3.Flow chart for Deep Learning SVM

IV. EXPERIMENTAL RESULTS

.NET FRAMEWORK

The .NET Framework has two main parts:

1. Common Language Runtime (CLR)
2. Hierarchical set of class libraries. The CLR is described as the "execution engine" of .NET. It provides the environment within which programs run. The most important features are,
3. Conversion from a low-level assembler –style language, called Intermediate Language (IL), into code native to the platform being executed on.
4. Memory management, notably including garbage collection.
5. Checking and enforcing security restrictions on the running code.
6. Loading and executing programs, with version control and other such features.

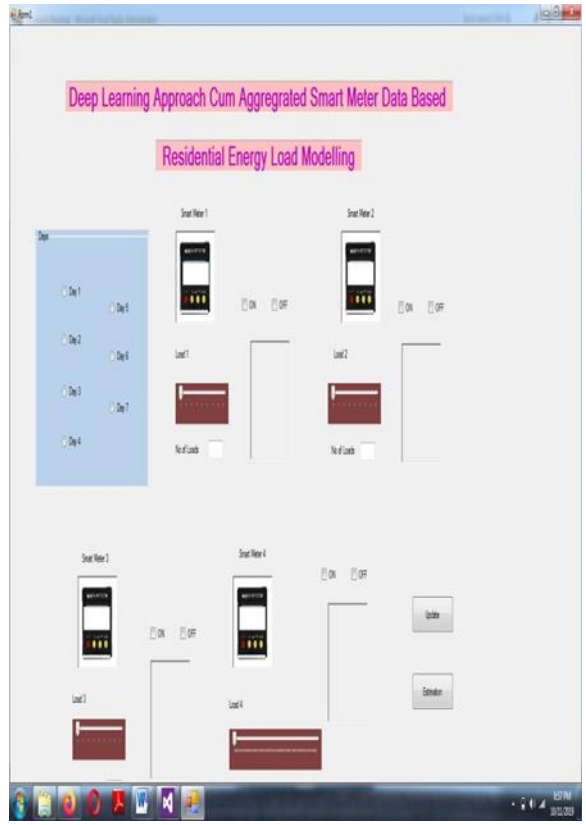

Fig. 4. Complete Model Design

Fig.5. Day Based Load Condition and Meter Operation

Fig.6. Data Updating and Data Management to Data Store

Fig.7. Estimation of Load and Metering Data on Day Basis

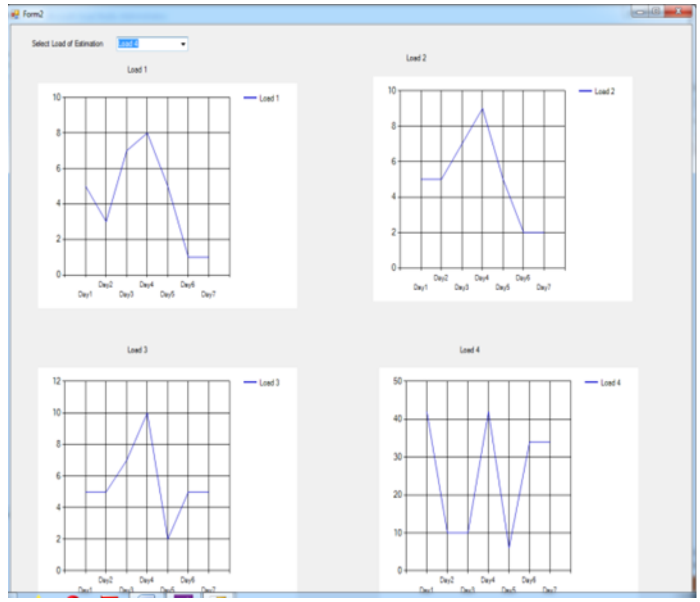

Fig.8. Day Based Load Estimation for a Week

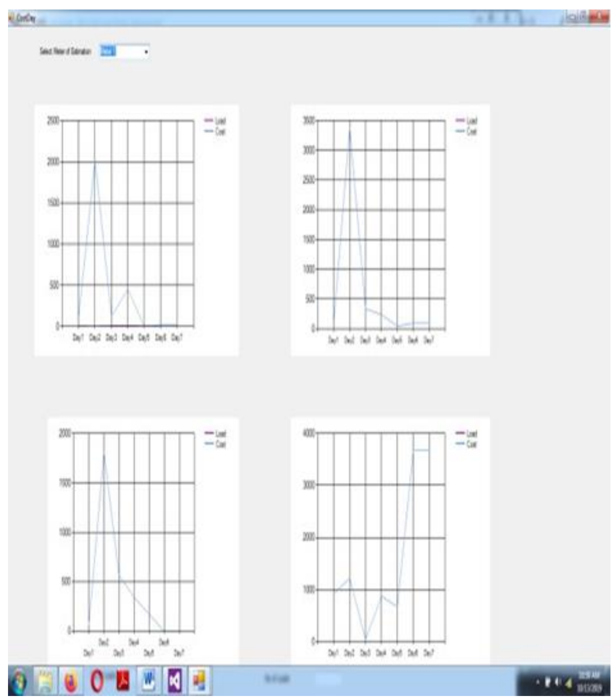

Fig.9. Load and Cost Estimation for a week

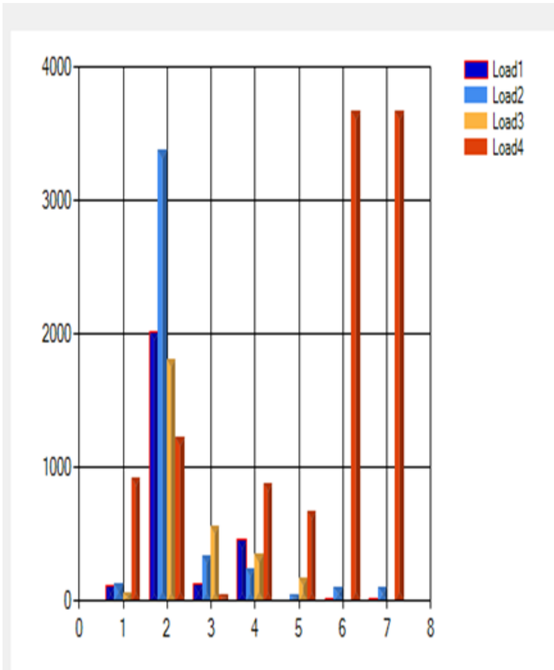

Fig..10 Load Based Estimation

V. CONCLUSION

This paper proposes a Deep learning technique for residential probabilistic load forecasting under high uncertainty and volatility was presented. A three-step pipeline that 1) clusters customers into groups, 2) pools force profiles of customers within each group, and 3) leverages these groups in a multitask learning framework, was proposed to generally share knowledge across groups while accounting for his or her differences. Numerical effects were presented which demonstrate that clustering-based pooling strategy can address the over fitting issue in deep learning. Further, the multi task learning framework provides superior forecasting performance with several popular benchmark methods. The proposed approach is extremely suited to practical applications such as for example residential demand response and network reliability analysis in smart grids.

REFERENCES

[1] Ashfaq Ahmad, Nadeem Javaid, Abdul Mateen, Muhammad Awais and Zahoor Ali Khan, Short-Term Load Forecasting in Smart Grids: An Intelligent Modular Approach, journal, Energies 2019, 12, 164.

[2] GaoXiuyun,WangYing,GaoYang,Sun,Chengzhi, X iangWen,YueYimiao, Short-term Load Forecasting Model of GRU Network Based on Deep Learning Framework; 978-1-5386-8549-5/18/$31.00 ©2018 IEEE

[3] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares,H. Schwenk, and Y. Bengio, “Learning phrase representations usingRNN encoder-decoder for statistical machine translation,” arXiv preprintarXiv:1406.1078, 2014.

[4] K. Grolinger, A. L’Heureux, M. A. Capretz, andL. Seewald, “Energy forecasting for event venues: Big data and prediction accuracy,” Energy and Buildings, vol. 112, pp. 222–233, 2016.

[5] K. Li, X. Xie, W. Xue, X. Dai, X. Chen, and X. Yang, “A hybrid teaching-learning artificial neural network for building electrical energy consumption prediction,” Energy and Buildings, vol. 174, pp. 323–334,2018.

[6] L. Tang, X. Wang, X. Wang, C. Shao, S. Liu, andS. Tian, “Long-term electricity consumption forecasting based on expert prediction and fuzzy bayesian theory,” Energy, vol. 167, pp. 1144–1154, 2019.

[7] M. Khairalla, X. Ning, N. AL-Jallad, and M. El-Faroug, “Short-term forecasting for energy consumption through stacking heterogeneous ensemble learning model,” Energies, vol. 11, no. 6, p. 1605, 2018.

[8] Pu Zhao1, Yanzhi Wang2, Naehyuck Chang3, Qi Zhu4, Xue Lin, A Deep Reinforcement Learning Framework for Optimizing Fuel Economy of Hybrid Electric Vehicles, 2018 IEEE

[9] Shunjiang Wang, Haitien Chen, BoamingPu, Hui Zhang, Shuangyaun Jin, Xinrui Liu, Analysis of Network Loss Energy Measurement Based on Machine Learning, International Conference on Information and Automation Wuyi Mountain, China, August 2018

[10] S. Han, Y.-h. Qiao, J. Yan, Y.-q. Liu, L. Li, andZ.Wang, “Mid-to-long term wind and photovoltaic power generation prediction based on copulafunction and long short term memory network,” Applied Energy, vol. 239,pp. 181–191, 2019.