Appliance-level Short-Term Load Forecasting using Deep Neural Networks

Ghulam Mohi Ud Din*, Andreas U. Mauthe†, and Angelos K. Marnerides†
* Department of Computer Science, Liverpool John Moores University, UK
g.mohiuddin@2015.ljmu.ac.uk
† InfoLab21, School of Computing & Communications, Lancaster University, UK
a.mauthe, angelos.marnerides@lancaster.ac.uk

***Abstract*—The recently employed demand-response (DR) model enabled by the transformation of the traditional power grid to the SmartGrid (SG) allows energy providers to have a clearer understanding of the energy utilisation of each individual household within their administrative domain. Nonetheless, the rapid growth of IoT-based domestic appliances within each household in conjunction with the varying and hard-to-predict customer-specific energy requirements is regarded as a challenge with respect to accurately profiling and forecasting the day-to-day or week-to-week appliance-level power consumption demand. Such a forecast is considered essential in order to compose a granular and accurate aggregate-level power consumption forecast for a given household, identify faulty appliances, and assess potential security and resilience issues both from an end-user as well as from an energy provider perspective. Therefore, in this paper we investigate techniques that enable this and propose the applicability of Deep Neural Networks (DNNs) for short-term appliance-level power profiling and forecasting. We demonstrate their superiority over the past heavily used Support Vector Machines (SVMs) in terms of prediction accuracy and computational performance with experiments conducted over real appliance-level dataset gathered in four residential households.**

***Index Terms*—Smart Grid, Power Forecasting, Deep Learning**

I. INTRODUCTION

Recently, governments of developed and developing countries have been introducing legislative procedures enforcing energy providers to install smart-meters in every residential or commercial building within their administrative domain. By 2020 the EU alone aims to replace 80% of traditional electricity meters with smart meters to support the objective of reducing greenhouse emissions up to 9% and further enabling a flexible and beneficial business model that optimally utilises energy resources for both themselves as well as their clientele [1]. These new Advanced Metering Infrastructures (AMIs) as composed by individual smart-meters encapsulate the main technological innovations in the general SmartGrid domain and act as the functioning block for the Demand-Response (DR) model in which energy demands are offered based on a client-explicit usage demand basis and in parallel provides proactive response in scenarios related to energy shortages or outages. Hence, the DR at the energy provider's side relies heavily on components that deal with the process of profiling and further forecasting the energy usage as reported by each smart-meter within each AMI.

Apart from profiling and forecasting the energy usage based on the aggregated measurements provided by smart-meters, it is also seen as far more accurate to characterise and forecast the usage patterns of individual appliances within a given household [2]. Further, enabled by the Internet of Things (IoT) technology, a modern household may be mapped as a Home Area Network (HAN) in which IoT appliances/devices (e.g. a washing machine) report their energy usage enabling a client to be fully aware of the consumption for each appliance in their household. Essentially, appliance-level short-term (i.e. day-to-day, week-to-week) load forecasting provides consumers the opportunity to map and manage their daily electricity usage profiles for each appliance, engage further in the DR business model through their smart-meters over the AMI and thus select suitable price plans. However, the task of accurately composing, profiling and forecasting models using such fine-grained and generally highly non-linear measurements are considered to be quite challenging and hard to confront especially on a short-term basis. In addition, the gathered measurements from each appliance are in many cases hard to predict since they are generated and related with exogenous factors such as weather conditions, consumer's social behavior, geographical locations, time of day, and day of the week. Hence, it is crucial to develop new profiling and forecasting methods that provide higher prediction accuracy through including and considering such factors.

Hence, the work introduced in this paper contributes by demonstrating the applicability of Deep Neural Networks (DNNs) for the purpose of appliance-level short-term power consumption profiling and forecasting. Due to their natural formulation as composed by multiple hidden layers, we find DNNs capable to adequately identify, relate and profile hidden structures between data features gathered by diverse data sources, thus providing accurate power consumption predictions in the context of our work. In more detail, using a Principal Component Analysis (PCA)-based feature selection scheme, we assess the efficiency of the Feed-Forward Deep Neural Network (FF-DNN) and the Recurrent Deep Neural Network (R-DNN) formulations in terms of prediction accuracy and computational performance over datasets gathered from appliances in four individual households. In addition, the paper compares the aforementioned formulations with the commonly used Support Vector Machine (SVM) algorithm and demonstrate their superiority. To the best to our knowledge we are the first to go beyond single-layer neural networks and demonstrate the applicability and benefits offered by DNNs in the context of appliance-level short-term power load forecasting.

The remainder of this paper is structured as follows: section II briefly discusses existing solutions available for short term load forecasting. Section III is dedicated at describing the algorithms used within this work whereas section IV presents the methodology employed in this research work. SectionV is dedicated to present the outcomes of our evaluation. Finally, section VI concludes and summarises this paper.

II. Background & Related Work

Power consumption profiling and forecasting can be divided into four main categories, namely: (i) very short term load forecasting (VSTLF), (ii) short term load forecasting (STLF), (iii) medium term load forecasting (MTLF) and (iv) long term load forecasting (LTLF). These categories represent the time granularity in which measurements are assessed (e.g. hour-to-hour, day-to-day, week-to-week, year-to-year, etc.) [1].

Recently, a number of studies have gone beyond traditional statistical forecasting schemes (e.g. as in [3], [4]) and adopt some of the benefits offered by machine learning (ML)-oriented techniques for STLF. Mainly, the direction towards ML solutions is seen as beneficial due to the ability of such approaches to model underlying patterns among several linear and non-linear data features gathered by heterogeneous inputs. AI-oriented techniques such as feed-forward neural networks (FF-NN), recurrent neural networks (RNN), radial basis functions (RBFs) and support vector machines (SVMs) are employed throughout current and past literature [1], [2], [5], [6]. The closest to our work is discussed in [2] in which the authors propose a multilayer FNN that in synergy with a wavelet-based analysis aids to predict the load of a given household based on various price signals as well as the power consumption measurements on individual appliances. The authors also introduce a comparison of the proposed scheme in terms of the obtained forecasting error with other formulations such as the traditional ANN and a hybrid linear model. Another interesting approach is introduced in [5] where an RNN-based model is proposed to formulate a prediction scheme for per-hour appliance-level power load forecasting in a given household.

Innovative, ML-based methodologies to improve the forecasting accuracy have also been proposed such as same day selection method, aggregating forecasting at different levels, feature selection method and weather station selection method [6]. Considering all these we argue that our work differs significantly since we are the first to go beyond single-layer neural networks. Our approach employs a pre- feature selection process[1] using PCA and further uses computationally efficient and accuracy-wise beneficial, multi-layer Deep Learning (DL) techniques to perform STLF by using appliance-level measurements.

[1]A process not performed in the studies we investigated.

III. Deep Neural Networks & Support Vector Machines

Traditional single-layer neural networks (e.g. ANN) which consist of one hidden layer, can model higher abstractions with minimal complexity. However, their successors that employ the ”deep” architecture (e.g. FF-DNN) and we use in this work, are capable at extracting granular and hidden structural representations due to their n-number of hidden layers at the cost of higher complexity and eventually higher processing time. The complexity and processing constraints are confronted by Hinton et al. [7] proposing the Deep Belief Network (DBN) where it is demonstrated that ”deep” architectures can be trained using a greedy layer approach. By virtue of their applicability in several domains and particularly in image processing, deep architectures are now considered as efficient in modelling non-stationary time series as it happens in the datasets used within this work.

A. Feed-forward Deep Neural Network (FF-DNN)

FF-DNN is a prototypical deep learning model, also called multi layer perceptron (MLP). As structurally inspired by the human neural network, the basic unit of FF-DNN, is a neuron. In humans, a signal with varying strength passes through synaptic junctions and aggregates at the neuron's activation as input. Similarly in the FF-DNN model, the inputs x are combined with weights w and biases b at the neuron using equation 1 as follows:

$$\phi = \sum_{n=0}^{N-1} w_n x_n + b \tag{1}$$

A nonlinear activation function σ is applied on computed ϕ which produces an output and sends as input to other connected neurons. The basic goal of FF-DNN is to approximate the function $\sigma(\phi)$.

FF-DNNs are made up of many neurons interconnected in different layers. The first layer of the network is the data input layer and represents the input feature space, in our case this is denoted as the feature set for each appliance as derived by the PCA feature selection scheme that we explain in section IV. The first layer in FF-DNN is then followed by multiple hidden layers accumulating non-linear activation functions. Finally, the last layer is the output layer and represents the output space based on either a classification or a regression problem. Given the nature of STLF we employed a regression problem in order to obtain a forecast estimation. In addition, we note that the output of the model is fully decided by weights and biases and the overall FF-DNN algorithm is able to learn by adjusting weights to minimize a given loss function in relationship with the pre-produced training dataset. The loss function is given by equation 2 as follows:

$$Loss(q|W,B) \tag{2}$$

where q represents each training example in the data, W is the matrix of weights and B is the set of biases.

B. Recurrent Deep Neural Network (R-DNN)

In FF-DNN, connections between neurons do not make cycles and information from input to output flows only in one direction. FF-DNN are powerful but lack in incorporating sequences and thus temporal patterns. R-DNN tackle this problem by utilising sequential information, thus R-DNN executes the same function on a given input recurrently and holds the outputs of previous iterations to be used in every new iteration. The multi-layer perceptron is the simplest form of R-DNN in which output of one hidden layer along with the input is fed back in the network.

The R-DNN can be seen as having $X(t)$ being the input at time t, $Q(t)$ the output, $Q(t-1)$ is input at time t, W_i is the weight for input layer, whereas W_{td} represents the weight for the time delay input and W_o is the weight for the output layer. Finally, f_1 and f_1 are hidden and output layer transfer functions. The time delay unit is required to hold the output and feed back at the next time step. The following non-linear equations delineate the functionality of the R-DNN's structure.

$$Q(t) = f_1(X(t) * W_i + Q(t-1) * W_{td}) \tag{3}$$

$$Y(t) = f_2(Q(t) * W_i) \tag{4}$$

C. Support Vector Machines (SVM)

The traditional two-class support vector machine (SVM) algorithm is a supervised learning model that does not belong into the DNN domain. This class of models is seen to be applicable for both classification and regression problems as demonstrated in [8]. The basic idea behind SVM is to map the original data nonlinearly into a higher dimensional feature space and then a linear model is applied to classify the given feature space, which in our case is the PCA output in the pre-processing stage we explain in section IV. The overall formulation of the SVM algorithm may be summarised in the following equation as:

$$f(x,w) = \sum_{i=1}^{n} w_i q_i(x) + b \tag{5}$$

where $q_i(x)$ represents the nonlinear transformation applied on the original input data and b is the bias term. The ϵ-insensitive loss function is used to estimate the correctness of nonlinear transformations defined as:

$$L_\epsilon(y, f(x,w)) = \begin{cases} 0, & \text{if } |y - f(x,w)| \le \epsilon; \\ |y - f(x,w)| - \epsilon, & \text{otherwise.} \end{cases} \tag{6}$$

The empirical risk of the function is calculated as:

$$R_{emp} = 1/n \sum_{i=1}^{n} L_\epsilon(y_i, f(x_i, w)) \tag{7}$$

In addition, the regularized risk function is minimized to estimate the parameters w and b.

$$R(C) = (C/N) \sum_{i=1}^{N} L(d_i, y_i) + w^2/2 \tag{8}$$

C, ϵ and kernel parameters influence the estimation accuracy of the SVM and their tuning is a significant stage within our experimentation.

TABLE I
DATASET MAIN FEATURES

Features			
Date	Time	Furnace HRV	Microwave
Dish Washer	Fridge	Dryer	Bedroom Lights
Temperature	Condition	Humidity	Visibility
Pressure	Wind Speed	Cloud Cover	Wind Bearing
Precipitation	Dew point	Precip Probability	Total Consumption

IV. METHODOLOGY

A. Data & Feature Selection

The datasets used in this work are appliance-level datasets and represent a year-long dataset for four individual houses obtained from the Smart* project [2] in the year 2016. As depicted in Table I each measurement sample contains the power consumption for each appliance alongside some environmental features. In order to achieve meaningful comparison between each house we perform some pre-processing since each house has differing appliances reported. Thus, within our experimentation we have selected the common appliances as depicted in Table I namely: furnace HRV, microwave, dish washer, fridge, dryer and bedroom lights.

Based on the datasets collection process reported in the Smart * project, the appliance-level power consumption values have been collected by sensors with varying measurement samples in each household (e.g. 1, 15, 30 minutes). Therefore, it is essential for our experimentation to normalise all data in order to ensure that they complied with the same sampling. For instance, the environmental measurements (e.g. humidity, fog, visibility etc.) is converted to the numerical values 1 to 10 using direct mapping and subsequently normalised under the min-max normalisation approach. The reason for applying normalisation is that large values in the datasets would essentially require large weights during the weight initialisation process in the employed DNNs, thus large biases would occur.

Within the preprocessing process we remove the features that have extremely small consumption values. These features include the consumption from various outlets and circuits in the house that can be considered as negligible based on empirical observations visualising the appliance level time-series. To further refine and extract prominent features the PCA technique is employed [9]. The principal components derived from the PCA algorithm can be assessed based on the variability between them. They illustrate the features that are mostly seen in the actual dataset; hence, the most dominant features. With the employment of PCA it is feasible to reduce the high-dimensionality of the examined datasets. Despite the fact that the original datasets require less complex models to produce some forecasting results it can be observed that the usefulness of the features on the post-PCA process increase

[2]Smart*: An Open Data Set and Tools for Enabling Research in Sustainable Homes: http://traces.cs.umass.edu/index.php/Smart/Smart

complexity but also improve the resulted prediction accuracy. Table II lists the most significant features extracted using PCA.

TABLE II
FEATURE SELECTION USING PCA

Features			
Date	Time	Furnace HRV	Microwave
Dish Washer	Fridge	Dryer	Bedroom Lights
Temperature	Humidity	Cloud Cover	Wind Bearing
Total Consumption			

B. Algorithm Parameter Selection

Subsequent to the PCA-based feature selection process, a critical step was to select and tune the hyper parameters for the estimators of the utilised FF-DNN, R-DNN and SVM algorithms that we briefly described earlier in section III. Naturally, the accuracy performance of these models relies heavily on the optimal selection of the hyper-parameters related to each algorithm/estimator. In order to achieve an optimal selection it was necessary to employ a trivial grid search technique which in practise considers all parameter combinations and chooses the best combination that provides the lowest prediction errors.

For instance, the FF-DNN's parameter selection process performed a grid search on activation functions, hidden layers, $l1$ and $l2$ regularization, learning rate, momentum, dropout ratio, number of epochs and epsilon. Table III depicts the outcome of the grid-based parameter search with the optimal values for each of the aforementioned activation functions.

TABLE III
FF-DNN OPTIMAL PARAMETERS

Parameters			
Training function	ReLU	Epochs	500
L1	0.0001	L2	0.0001
Learning rate	0.005	Epsilon	0.0001
Dropout ratio	0.1	Hidden layer	[10 10]

By contrast with the FF-DNN algorithm, the R-DNN has cyclic connections and delays between layers and the nodes that behave as internal memory and assist the algorithm to remember previous computations. Though feedback connections increase the convergence time and require more resources for computations, this eventually helps in getting accurate predictions with least errors. Nonetheless, the majority of hyper parameters in R-DNN are similar with those in the FF-DNN and through the grid search technique it was feasible to obtain the most optimal values as shown in table IV.

TABLE IV
R-DNN OPTIMAL PARAMETERS

Parameters			
Training function	trainlm	Epochs	200
L1	0.0001	Hidden layer	[5 5 5]
Learning rate	0.01	Dropout ratio	0.1

The SVM model's prediction performance depends on the gamma and C parameters as well the chosen kernel function. In our case it was derived that the Radial Basis Function (RBF) was the most suitable and accuracy-wise efficient since several trials with the other functions (e.g. gaussian, sigmoid) produced erroneous outputs with high processing requirements. Similarly with the parameter selection process in FF-DNN and R-DNN we applied the grid search technique and derived the most optimal values for gamma to be equal to 1.0 and C to be equal to 0.5.

V. EVALUATION

The proposed models are evaluated on the basis of three error metrics; the mean squared error (MSE), the root mean squared error (RMSE) and the mean absolute error (MAPE). These metrics are selected due to their wide use in several studies (e.g. as in [8]) and their joint view provides a holistic viewpoint with respect to the prediction performance of the examined algorithms. The MSE metric allows to calculate mean errors among actual measurements and predictions, however it cannot compensate the effects of outliers in the error distributions. Thus, we address this problem by calculating the RMSE that considers the squares of difference between actual and prediction values and captures the effects of lower and higher consumption values. Finally, we choose to use the MAPE metric since it provides a percentage-based error visualisation. In addition, we use appliance-level data gathered for 2 weeks in order to compose a training model for each algorithm and choose new testing data from a random day of the week and perform a prediction for that particular day.

As evidenced in table V we map the RMSE and the MSE metrics to indicate the normalised loss of KW/h for a given day forecast as being the main objective of any STLF application. In parallel, we also assess the computational performance with respect to the time taken for each algorithm to produce a prediction. Based on the outcomes of our experimentation, we witness in all forecasts, that both DNNs outperform the SVM formulation in most of cases. In particular the FF-DNN algorithm is the most computationally optimal for providing quick predictions. For instance, by closely observing the outcomes for House A, the R-DNN holds the lowest MSE from all three evaluated algorithms however it has comparably a very close value on the RMSE metric with FF-DNN and same MAPE on a percentage basis. On the other hand, the SVM formulation seems to perform better than FF-DNN on the MSE metric, nonetheless extremely worse in the RMSE and MAPE metrics. Also, by comparing all three algorithms for House A it is also evidenced that the most computationally intensive model is the R-DNN followed by SVM.

The observations for House A are quite similar for the rest three residential households and surely there is a trade-off between accuracy and computational performance. We argue that these outcomes are produced due to the intrinsic formulations of each algorithm individually. The high computational cost but the high accuracy performance obtained for R-DNN relates to the fact that the R-DNN relies on two inputs at times t and $t-1$ due to the cyclic connections between nodes and layers. Thus, it requires time to store/access information

TABLE V
SUMMARY FOR DAY AHEAD STLF PREDICTION ERRORS AND COMPUTATIONAL TIME FOR FF-DNN, R-DNN AND SVM

House	Model	MAPE(%)	RMSE(KW/h)	MSE(KW/h)	Time(sec)
House A	SVM	0.32	0.21	1.96	133
	FF-DNN	0.01	0.008	11.7	1.37
	R-DNN	0.01	0.004	1.49	960
House B	SVM	0.26	0.23	3.88	215
	FF-DNN	0.003	0.002	4.35	1.40
	R-DNN	0.017	0.0023	11.7	995
House C	SVM	0.37	0.28	2.51	223
	FF-DNN	0.003	0.002	3.39	1.44
	R-DNN	0.020	0.0034	1.66	985
House D	SVM	0.93	0.37	2.55	200
	FF-DNN	0.13	0.03	3.62	2.64
	R-DNN	0.0012	0.0002	0.90	979

and repetitively compare any newly tested instance with its nodes on each layer. In parallel, the SVM formulation spends a lot of effort on constructing the high-dimensional vectors and it seems to perform worse than DNNs in general since its marginal boundaries cannot adequately capture the high variations of power usage on the appliance-level measurements. On the other hand, the FF-DNN algorithm performs reasonably better in terms of both the accuracy performance as well as the minimal computational time required. FF-DNNs do not need to store/access information as in R-DNNs and their corresponding weight initialization for nodes and layers is performed relatively quickly, hence providing a prediction much quicker than the R-DNN but with slightly less accuracy.

VI. CONCLUSIONS

Short-term Load Forecasting (STLF) is an important task in the context of optimal energy management for both the energy provider as well as the individual client. A granular and accurate approach for composing STLF is to consider the measurements gathered by each appliance in a given household. Nonetheless, the dynamic, customer-specific behaviour of power utilisation over different appliances as well as their indirect relationships with exogenous features (e.g. weather, humidity, etc) pose a challenge to any STLF scheme and a method to capture these properties is required. Therefore, in this work we go beyond traditional forecasting schemes and we propose the applicability of Deep Neural Networks (DNNs) for the STLF in individual households by using appliance-level measurements. We propose a methodology in which a critical step is a PCA-based feature selection process is present and we subsequently assess and compare 2 DNN-based approaches, the FF-DNN and the R-DNN, with the commonly used SVM formulation for day-ahead forecasting using real data from 4 individual households. Our results demonstrate the superiority of the DNN-based solutions due to their ability to adequately relate the actual appliance-level power measurements with exogenous features with reasonably acceptable trade-offs between accuracy and performance for the FF-DNN formulation in particular. We argue that our results establish a concrete basis for further exploitation of DNNs in the context of power consumption profiling for improving future energy management systems and also broadening the avenues for domains such as appliance-level fault diagnosis and security.

ACKNOWLEDGMENTS

The authors would like to thank the UK EPSRC funded ”Situation Aware Information Infrastructure” (SAI^2 - EP/L026015/1) and the ”Towards Ultimate Convergence of All Networks” (TOUCAN - EP/L020009/1) projects that have kindly supported this work.

REFERENCES

[1] G. M. U. Din and A. K. Marnerides, “Short term power load forecasting using deep neural networks,” in *Computing, Networking and Communications (ICNC), 2017 International Conference on*. IEEE, 2017.

[2] N. G. Paterakis, A. Tascikaraoglu, O. Erdinc, A. G. Bakirtzis, and J. P. S. Catalao, “Assessment of Demand-Response-Driven Load Pattern Elasticity Using a Combined Approach for Smart Households,” *IEEE Transactions on Industrial Informatics*, vol. 12, no. 4, pp. 1529–1539, 2016.

[3] A. Kavousi-Fard and F. Kavousi-Fard, “A new hybrid correction method for short-term load forecasting based on arima, svr and csa,” *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 25, no. 4, pp. 559–574, 2013.

[4] S. Dwijayanti and M. Hagan, “Short term load forecasting using a neural network based time series approach,” in *2013 1st International Conference on Artificial Intelligence, Modelling and Simulation*, Dec 2013, pp. 17–22.

[5] A. Marvuglia and A. Messineo, “Using recurrent artificial neural networks to forecast household electricity consumption,” *Energy Procedia*, vol. 14, pp. 45–55, 2012.

[6] T. Hong and S. Fan, “Probabilistic electric load forecasting: A tutorial review,” *International Journal of Forecasting*, vol. 32, no. 3, pp. 914–938, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.ijforecast.2015.11.011

[7] G. Hinton, Y. Bengio, and Y. Lecun, “Breakthrough Deep Learning : machine learning algorithms based on,” 2015.

[8] A. Marnerides, S. Malinowski, R. Morla, and H. Kim, “Fault diagnosis in dsl networks using support vector machines,” *Computer Communications*, vol. 62, pp. 72 – 84, 2015.

[9] J. E. Jackson and G. S. Mudholkar, “Control procedures for residuals associated with principal component analysis,” *Technometrics*, vol. 21, no. 3, pp. 341–349, 1979.