



Research paper

An enhanced CNN-LSTM based multi-stage framework for PV and load short-term forecasting: DSO scenarios

Mohammad Ahmad A. Al-Ja'afreh ^{a,*}, Geev Mokryani ^a, Bilal Amjad ^b^a Faculty of Engineering and Informatics, University of Bradford, Bradford, BD7 1DB, UK^b Northern Powergrid, Aketon Road, Castleford, West Yorkshire, WF10 5DS, UK

ARTICLE INFO

Article history:

Received 18 November 2022

Received in revised form 22 June 2023

Accepted 1 August 2023

Available online 18 August 2023

Dataset link: <https://data.mendeley.com/datasets/zympd537wv/1>

Keywords:

CNN-LSTM

Feature generation

PV short-term forecasting

Load short-term forecasting

Distribution system operator

ABSTRACT

The importance of accurate forecasting in the electric sector has grown due to the increasing demand and adoption of high volume of Renewable Energy Sources (RES). Short-term forecasting (STF) using deep learning methods has shown potential for improving forecasting accuracy. However, the accuracy of these methods can be further enhanced by combining them to generate a hybrid model, selecting appropriate input features, generating new features, and optimizing model parameters. This paper proposes a novel multi-stage framework for PV and load STF that employs feature generation, feature selection, and optimal hyperparameter tuning preprocessing techniques. An enhanced hybrid CNN-LSTM deep learning model architecture is developed in the final stage of the proposed framework. The framework is assessed and compared to other leading-edge approaches across different DSO scenarios, including multiple single-phase residential loads, three-phase feeders, and secondary substation, demonstrating a significant reduction in forecasting errors.

© 2023 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Motivation and background

In the race towards a net-zero future, electrical power systems are moving from centralized to decentralized operation. Instead of having large fossil fuel power plants connected to transmission networks only, the electrical grid is experiencing a growing number of distributed energy resources connected to distribution networks. Moreover, an increasing number of prosumers, electric vehicle users, energy storage systems, heat pumps, and other low-carbon technologies are making distribution networks more complex to operate (Nijhuis et al., 2015). To meet these challenges, distribution network operators are transitioning towards distribution system operators (DSO), aiming to monitor actively, control, and optimize the network smartly and efficiently at the local level (Boyd, 2017). The local generation and load demand short-term forecasting (STF) will play a vital role in active network management, optimal load dispatching, and local markets within local energy systems (Nijhuis et al., 2015). The forecasting purpose and its application varied based on the grid operation level. For instance, at the residential level, Photovoltaic (PV) system

generation and load forecasts enable prosumers to actively participate in the local energy market by exporting excess energy and participating in demand response and vehicle-to-grid programs. Feeder-level forecasts can help DSO to control and optimize technical network parameters such as losses and node voltage at low voltage (LV) networks (Haben et al., 2021; Sorour et al., 2021). Furthermore, substation level forecasts would be helpful for grid balancing and effectively enable flexibility services. Therefore, this study proposed a multi-stage STF framework for various DSO scenarios, including load STF for residential, three-phase feeders, and secondary substation demand and STF for solar PV generation.

1.2. Review of related works

Solar PV short-term forecasting (PV-STF) and load short-term forecasting (L-STF) techniques are widely addressed in state-of-the-art. The most commonly used STF methods can be classified into two main categories: statistical methods (Bennett et al., 2014; Goude et al., 2013; Takeda et al., 2016; Wang et al., 2018; Zhang et al., 2018), and artificial intelligence (AI) methods (Aouad et al., 2022; Aprillia et al., 2020; Budin et al., 2022; Dai et al., 2022; du Plessis et al., 2021; Huang and Kuo, 2019; Jahangir et al., 2020; Kabilan et al., 2021; Kim and Cho, 2019; Korkmaz et al., 2021; Laouafi et al., 2017; Li et al., 2018, 2020; Ma et al., 2021; Mishra et al., 2020; Motepe et al., 2019; Parvez et al., 2020; Razavi et al., 2020; Sajjad et al., 2020; Shi et al., 2017; Tan et al., 2022;

* Corresponding author.

E-mail addresses: [M.A. Al-Ja'afreh](mailto:M.A.Al-JaAfreh@brad.ac.uk), [G. Mokryani](mailto>G.Mokryani@brad.ac.uk), bilal.amjad@northernpowergrid.com (B. Amjad).

Nomenclature	
AI	Artificial intelligence
ANFIS	Adaptive neuro-fuzzy inference system
ANN	Artificial neural networks
ARIMA	Autoregressive integrated moving average
BLSTM	Bidirectional long-short term memory
CNN	Convolutional neural networks
DL	Deep learning
DSO	Distribution system operator
ECNN-LSTM	Enhanced CNN-LSTM
ELM	Extreme learning machine
FANN	Fuzzy ARMAP neural network
FFNN	Feedforward neural network
FG	Feature generation
FS	Feature selection
GAM	Generalized additive model
GRU	Gated recurrent units
KF	Kalman filtering
L-STF	Load short-term forecasting
LSTM	Long-short term memory
LV	Low voltage
MAE	Mean absolute error
MAPE	The mean absolute percent error
MISO	Multi-input single output
ML	Machine learning
MLP	Multi-layer perceptron
MSE	Mean squared error
OHPT	Optimal hyper parameter tuning
PV	Photovoltaics
PV-STF	PV short-term forecasting
RF	Random forest
RMSE	Root means square error
RNN	Recurrent neural networks
SSA	Slap swarm algorithm
STF	Short-term forecasting
SVM	Support vector machines
TPE	Tree Parzen estimator
WPD	Wavelet packet decomposition
WT	Wavelet transform
$p_{app,i}^t$	Load profile for each appliance
$P_{customer}^t$	Customer-level load profile
N	Number of appliances
X	The original feature value
X_{min}	The minimum value of that feature
X_{max}	The maximum value of that feature
X_{scaled}	The normalized value
$f_1 - f_5$	Newly generated features
t_f	Target feature
i	Index of the data point in the data set
n	Number of data points at every hour
x_i	The x variable sample
y_i	The y variable sample
\bar{x}	The mean of values in x variable
\bar{y}	The mean of values in y variable
x_j^l	The output value of j th neuron in the feature map at the l th layer
x_i^{l-1}	The input feature map at $(l-1)$
w_{ij}^l	Is the convolutional filter connected the i th input feature map at $l-1$ with j th feature map at l
b_j^l	Convolutional neural network bias
$sigmoid$	Sigmoid activation function
$tanh$	Hyperbolic tanh activation function
\otimes	Elementwise multiplication
f_t, i_t, o_t	Forget, input and output gates
x_t	Current input data vector
c_t	Current state cell
h_{t-i}	Previous hidden state
c_{t-1}	Previous cell state
w_{xf}	The weighting vector of the forget layer that are connected to input vector x_t
w_{hf}	The weighting vector of the forget layer that are connected to the previous hidden state h_{t-i}
w_{xi}	The weighting vector of the input gate layer that are connected to the input vector x_t
w_{hi}	The weighting vector of the input gate layer that are connected to previous hidden state h_{t-i}
w_{xo}	The weighting vectors for the output gate layer that are connected to the input vector x_t
w_{ho}	The weighting vectors for the output gate layer that are connected to the previous hidden state h_{t-i}
b_f, b_i, b_o	The associated bias to the forget, input, and output gates layers
C and ε	SVM predetermined model parameters
ξ_i and ξ_i^*	SVM model slack variables account for the deviation between the predicted and actual values
α_i and α_i^*	Lagrange multiplier
y_t	Current value being predicted
u_t	ARIMA model white noise or error term
μ	The expectation of y_t .
$a_1 \dots a_p$	Autoregressive parameter
$m_1 \dots m_q$	Moving average parameter
y_i	Actual value vector
\hat{y}_i	Predicted value vector
n_t	Number points in testing data set

The most common statistical methods used for PV-STF and L-STF are autoregressive integrated moving average (ARIMA) (Bennett et al., 2014; Wang et al., 2018; Zhang et al., 2018), Kalman filter (KF) (Takeda et al., 2016), and Generalized Additive Model (GAM) (Goude et al., 2013). Several studies have applied the aforementioned statistical methods. For instance, Ref. Zhang et al. (2018) proposed a hybrid model for short-term electricity load forecasting. The hybrid model combines different techniques, including improved empirical mode decomposition, ARIMA, and a wavelet neural network. The simulation results showed that the hybrid model performs well in electricity load forecasting than support vector machines (SVM), Elman neural network, and

extreme learning machine (ELM). Ref. Takeda et al. (2016) developed a framework for modeling electricity loads, and it was stated that this framework could be used for both forecasting and analysis. It combines the ensemble KF technique with shrinkage/multiple regression methods. Firstly, a state-space model is employed to model the load structure, the KF is used for the estimation, and the shrinkage/multiple linear regression method is applied to further improve the accuracy. The results illustrated that the forecasting accuracy of the proposed models is higher than that of the current state-of-the-art models. Ref. Goude et al. (2013) investigated a semi-parametric approach based on GAM theory to model electrical load over more than 2200 substations of the French distribution network. The GAM estimates the relationship between the explanatory variables and load: calendar variables, temperatures, etc. It was stated that the forecast performance achieved is relatively good compared to existing methods and can be computed at a low computational cost. Ref. Bennett et al. (2014) presented a model to forecast three-phase LV substation load demand. Although the proposed method classifies input variables using discrete artificial neural network (ANN) classification, the authors utilize ARIMA statistical method to perform the forecasting. This technique failed for residential L-STF due to the complex nature of the data, as the sudden load variations are more frequent in residential loads. Therefore, statistical models are inherently incompatible with capturing non-linear and complex relationships between features. Moreover, it was reported by Ref. Aouad et al. (2022) that the statistical methods fail to model load demand and PV generation due to two shortcomings. Firstly, it only uses previous electric load/PV generation inputs, while not utilizing the other essential factors including weather, calendar features, and special events. Secondly, statistical methods fail to model the nonlinear relationship between the lagged values of the input time series, and hence provide inaccurate predictions. For instance, in the winter or cloudy/rainy weather, the PV data is complex as the variation becomes more frequent and unexpected. In addition, in case of L-STF when moving from aggregated substation load profile at medium voltage level to residential load profile at low voltage level, the load profile become more complex and the relationship between feature become highly non-linear, due to sudden changes in the demand pattern of individual households which depend on the human behaviors and lifestyle (Aouad et al., 2022).

The second category includes AI methods, and generally, it can be classified into two main groups, machine learning (ML) (Kabilan et al., 2021; Laouafi et al., 2017; Li et al., 2018; Ma et al., 2021; Parvez et al., 2020; Valgaev et al., 2017; VanDeventer et al., 2019; Yang et al., 2019) and deep learning (DL) (Aprillia et al., 2020; Budin et al., 2022; Dai et al., 2022; du Plessis et al., 2021; Huang and Kuo, 2019; Jahangir et al., 2020; Kim and Cho, 2019; Korkmaz et al., 2021; Li et al., 2020; Mishra et al., 2020; Motepe et al., 2019; Razavi et al., 2020; Sajjad et al., 2020; Shi et al., 2017; Tan et al., 2022; Trivedi et al., 2022; Zang et al., 2018). Traditional ML methods have developed, such as random forest (RF) (Li et al., 2018) and ELM (Ma et al., 2021), Multi-layer perceptron (MLP) (Parvez et al., 2020), and SVM (Laouafi et al., 2017; VanDeventer et al., 2019). Several studies have applied ML methods for PV-STF and L-STF. For instance, Ref. Yang et al. (2019) developed a hybrid intelligent model based on a combination of autocorrelation feature selection and least squares support vector machine for L-STF. The autocorrelation function is employed to select the optimal input variables, and the least squares support vector machine parameters are optimized by the grey wolf Optimization algorithm. The results showed that the developed method could successfully forecast short-term electricity load. Ref. Valgaev et al. (2017) applied the K-Nearest Neighbors method to forecast the daily load curves of three different smart buildings. However,

it was reported that the applied forecaster K-Nearest Neighbors method could be further improved by explicitly considering PV generation, electrical heating, and ventilation systems installed in the building. Ref. Kabilan et al. (2021) applied ANN, SVM, decision tree, and multi-variable regression for domestic PV-STF and proved that ANN is the most accurate among these three methods. However, the forecasting accuracy of the existing ML methods relies entirely on the quality of the extracted features. Hence it requires prior knowledge and expertise to extract and select informative features from the dataset. Moreover, It was claimed by Ref. Kim and Cho (2019) that the existing ML methods have severe overfitting when the correlation of variables becomes complicated or the amount of data increases.

A branch of AI called DL has achieved remarkable success in various applications, including image processing, speech recognition, etc., and it has been recently applied for STF application due to comparatively higher forecasting accuracy. Several DL methods have been developed, such as convolutional neural network (CNN) (Aprillia et al., 2020; Huang and Kuo, 2019; Korkmaz et al., 2021), Gated Recurrent Unit (GRU) (Dai et al., 2022), recurrent neural network (RNN) (Shi et al., 2017), long short-term memory (LSTM) (Razavi et al., 2020), gated temporal CNN (Tan et al., 2022), and bidirectional LSTM (Jahangir et al., 2020). Numerous studies have applied DL methods for PV-STF and L-STF. For instance, Ref. Aprillia et al. (2020) proposed a short-term PV power forecasting strategy based on CNN and slap swarm algorithm (SSA). In this study, historical data were used as input variables containing weather information and classified into five weather types, such as rainy, heavy cloudy, cloudy, light cloudy, and sunny. The CNN method is used as a classification to forecast the next day's weather type. The CNN method is optimized using SSA to identify the optimal parameters of the CNN model. The proposed CNN-SSA was evaluated on a PV power generation system, and the results showed that the CNN-SSA method achieves better forecasting accuracy than SVM-SSA and LSTM-SSA methods. Ref. Shi et al. (2017) developed a novel pooling-based deep RNN to address the overfitting challenges brought by increasing the number of deep neural network layers. The main aim of this paper is to explore the potential of utilizing the DL method for household L-STF under high uncertainty and volatility. The results illustrated that the developed pooling based deep RNN method outperforms the existing benchmark, such as ARIMA by 19.5%, support vector regression by 13.1% and classical deep RNN by 6.5% in terms of root mean squared error (RMSE). Ref. Razavi et al. (2020) developed short-term net energy forecasting based on multi-input single-output model and LSTM. It considers the spatial relationships of various households' profile indirectly. The load profiles of all households are fed into the LSTM model, while either the aggregate profile or only one household is the target of forecasting. The results illustrated that LSTM is more resilient to sudden changes at the single household level, on the other hand, the developed Multi input single output LSTM based method is efficient for aggregate level. Ref. Tan et al. (2022) proposed a novel load forecasting based on deep multi-task learning method for higher quality multi-node load forecasting problem. In this study, the multi-modal feature extraction module is employed to extract features of different nodes from load data, and then the multi-objective deep neural network based on soft sharing multi-task learning is developed to improve the forecasting accuracy of the method. The results showed that the proposed forecasting method achieves the lowest forecasting error in real-world datasets compared to LSTM, ELM, reverse training based Fuzzy ARMAP neural network (FANN), Fuzzy ARTMAP and FANN. When overfitting occurs, it is difficult to predict long-term consumption. Other published work that made a comparison between various DL methods can be found in Refs. Trivedi et al. (2022) and du Plessis et al. (2021).

Table 1

Comparison of presented work with the state-of-art methods.

Models	FS	FG	OHPT	PV-STF	L-STF	3-Phase	Application
ARIMA , SVM (Wang et al., 2018)	x	x	x	✓	x	x	PV plant
ARIMA (Zhang et al., 2018)	x	x	x	x	✓	x	Regional load (MW)
KF (Takeda et al., 2016)	x	x	x	x	✓	x	Regional load (GW)
GAM (Goude et al., 2013)	x	x	x	x	✓	x	Substation (kw)
ANN-ARIMA (Bennett et al., 2014)	x	x	x	x	✓	x	Substation(kw)
CNN-Sequence-to-Sequence (Aouad et al., 2022)	x	x	x	x	✓	x	Residential
RF (Li et al., 2018)	x	x	x	x	✓	x	Residential
ELM (Ma et al., 2021)	x	x	✓	✓	x	x	PV plant (kw)
MLP (Parvez et al., 2020)	x	x	x	✓	x	x	Residential rooftop PV
Genetic algorithm based SVM (VanDeventer et al., 2019)	x	x	✓	✓	x	x	Commercial rooftop PV
SVM (Lauoui et al., 2017)	x	x	x	x	✓	x	Regional load (GW)
SVM (Yang et al., 2019)	✓	x	✓	x	✓	x	Regional load (GW)
K-Nearest (Valgaev et al., 2017)	x	x	x	x	✓	x	Three commercial buildings
SVM (Kabilan et al., 2021)	x	x	x	✓	x	x	Residential Building PV
SSA based CNN (Aprillia et al., 2020)	x	x	✓	✓	x	x	PV power plant
CNN (Huang and Kuo, 2019)	x	x	x	✓	x	x	PV power plant
CNN (Korkmaz et al., 2021)	x	x	x	✓	x	x	Large PV power plant
GRU (Dai et al., 2022)	✓	x	✓	✓	x	x	PV power plant
RNN (Shi et al., 2017)	x	x	x	x	✓	x	Residential/Commercial
LSTM (Razavi et al., 2020)	x	x	x	✓	✓	x	Residential/Commercial
GTCN-GRU (Tan et al., 2022)	✓	✓	x	x	✓	x	Regional load (MW)
BLSTM (Jahangir et al., 2020)	x	x	x	x	✓	x	Substation
GRU, LSTM, RNN, CNN (Trivedi et al., 2022)	✓	✓	✓	✓	✓	x	Residential/Commercial
FFNN , LSTM , GRU (du Plessis et al., 2021)	x	x	✓	✓	x	x	PV power plant
WPD-LSTM (Li et al., 2020)	x	x	x	✓	x	✓	Residential/Commercial
CNN-LSTM (Kim and Cho, 2019)	✓	x	x	x	✓	x	Residential/Commercial
BLSTM-RNN, LSTM , ELM, ANFIS (Motepe et al., 2019)	x	x	✓	x	✓	x	Substation
WT-LSTM (Mishra et al., 2020)	✓	x	x	✓	x	x	PV power plant
VMD-CNN (Zang et al., 2018)	x	x	x	✓	x	x	PV power plant
MLP, LSTM, BLSTM, CNN, CNN-LSTM, CNN-BLSTM (Budin et al., 2022)	x	x	✓	x	✓	x	LV Feeder
CNN-GRU (Sajjad et al., 2020)	✓	x	x	x	✓	x	Residential/Commercial
Proposed	✓	✓	✓	✓	✓	✓	DSO scenarios

Although reasonable results have been obtained from the application of DL for PV-STF and L-STF, the forecasting accuracy of DL methods can be further improved by combining two DL methods generating hybrid methods, selecting appropriate input features, generating new features from the input data, and optimizing the forecasting model parameters. Several studies have applied hybrid DL methods for PV-STF and L-STF. For instance, in Ref. Li et al. (2020), a hybrid DL-based wavelet packet decomposition (WPD) and LSTM were developed for short-term PV power forecasting. The hybrid DL method utilizes one-hour-ahead PV power forecasting with a five-minute interval. In this study, the WPD is employed to decrease the PV power data series into several sub-series, then four independent LSTMs are proposed for these sub-series. Next, the outputs predicted by each LSTM are reconstructed, and a linear weighting method is applied to get the final forecasting outputs. The results demonstrated that the developed hybrid DL method based on WPD and LSTM exhibits superior performance compared to a single LSTM, RNN GRU, in forecasting accuracy and stability. Ref. Kim and Cho (2019) proposed a hybrid DL method based on CNN-LSTM to estimate the electric energy consumption in real residential houses. The CNN method is employed to extract the representative features among several variables that affect energy consumption prediction. Then the output is fed into the LSTM for modeling temporal information of irregular trends in time series components. The results showed that the proposed hybrid CNN-LSTM method achieves the best prediction performance for electric energy consumption compared to LSTM, GRU, and BLSTM. However, it was stated that the performance of the CNN-LSTM model could be improved by determining the optimal hyper parameter tuning using an optimization algorithm as the trial-and-error criteria are considered ineffective in determining the optimal hyper parameter. Other Hybrid DL methods were proposed by Refs. Budin et al. (2022), Li et al. (2020), Mishra et al. (2020), Motepe et al. (2019), Zang et al. (2018) and Sajjad et al. (2020). For instance, Ref. Budin

et al. (2022) performed day ahead load STF for 12 aggregated households and did the comparison between DL (MLP, CNN, LSTM, and BLSTM) and hybrid DL (CNN-LSTM and CNN- BLSTM). It was stated that the CNN-BLSTM showed great opportunity for the time series forecasting problems. However, further investigation needs to be carried out to improve the forecasting accuracy of the model. In addition, none of the reviewed studies examine the developed model for both PV and L-STF at each voltage level, including medium and low voltage.

Furthermore, feature generation (FG), feature selection (FS), and optimal hyperparameter tuning (OHPT) are various techniques that have been proved to improve the accuracy of the forecasting model. For instance, Ref. Trivedi et al. (2022) investigated the impact of data generalization, FG and FS, and OHPT on GRU, RNN, CNN, and LSTM and proved the improvement in the accuracy of forecasting models. However, Refs. Aprillia et al. (2020) and Zang et al. (2018) used variational mode decomposition and slap swarm optimization, respectively, to optimize CNN forecasting model parameters and proved an enhancement in the overall accuracy of the forecasting model.

To the best of the author's knowledge, none of the existing literature addressed the following: (a) utilized two leading-edge DL models, namely CNN and LSTM, to propose a novel enhanced CNN-LSTM hybrid DL model for PV and load STF; (b) combines three preprocessing techniques, namely FG, FS, and OHPT; (c) covered various DSO forecasting scenarios, including multiple individual residential load forecasting, three-phase LV feeders load forecasting, three-phase secondary substation load forecasting, and residential solar PV generation forecasting.

Table 1 provides a high-level comparison between the proposed and state-of-the-art methods. For instance, Ref. Trivedi et al. (2022) has considered the application of PV and L-STF based on a combination of FG, FS, and OHPT for only single-phase residential load demand. However, it ignored three-phase substation and LV feeders load demand forecasting and hybrid DL

model architecture. Refs. Dai et al. (2022), Kim and Cho (2019), Mishra et al. (2020) and Sajjad et al. (2020) considered FS but ignored FG and OHPT. Similarly, Refs. Aprillia et al. (2020), du Plessis et al. (2021), Ma et al. (2021), VanDeventer et al. (2019) and Motepe et al. (2019) considered OHPT but failed to include FS and FG. Moreover, most of the existing studies either focused on PV-STF (Aprillia et al., 2020; Dai et al., 2022; du Plessis et al., 2021; Huang and Kuo, 2019; Kabilan et al., 2021; Korkmaz et al., 2021; Li et al., 2020; Ma et al., 2021; Mishra et al., 2020; Parvez et al., 2020; VanDeventer et al., 2019; Wang et al., 2018; Zang et al., 2018), or L-STF (Aouad et al., 2022; Bennett et al., 2014; Budin et al., 2022; Goude et al., 2013; Jahangir et al., 2020; Kim and Cho, 2019; Laouafi et al., 2017; Li et al., 2018; Ma et al., 2021; Mishra et al., 2020; Motepe et al., 2019; Sajjad et al., 2020; Shi et al., 2017; Takeda et al., 2016; Tan et al., 2022; Valgaev et al., 2017; Yang et al., 2019; Zhang et al., 2018). On the other hand, few studies have implemented L-STF at LV feeders (Budin et al., 2022) and substation levels (Bennett et al., 2014; Goude et al., 2013; Jahangir et al., 2020; Motepe et al., 2019).

However, the proposed framework combines three preprocessing techniques, namely FG, FS, and OHPT, for both PV and L-STF on various DSO scenarios. The developed framework is verified through the experimental case study. Furthermore, these preprocessing techniques are employed with the developed enhanced hybrid CNN-LSTM architecture, which has not been addressed in the existing literature to the best of the authors' knowledge.

1.3. Contributions and approach

The contribution of this paper is summarized as follows:

1. A novel FG technique is proposed where a set of equations is designed to generate additional features from the available historical data. The developed FG aims to reduce the non-linear relationship between input features, improving the DL model learning ability and forecasting accuracy. Then, the Pearson correlation technique is applied for FS to select the most relevant input features and eliminate irrelevant ones. Furthermore, the Tree Parzen Estimator (TPE) algorithm finds the optimal hyperparameter for forecasting models.
2. An enhanced hybrid CNN-LSTM structure is developed for PV-STF and L-STF and applied to different DSO scenarios. DSO scenarios include multiple residential loads, three-phase LV feeders load, three-phase secondary substation load, and solar PV generation. The developed framework is verified via various forecasting exercises for the listed scenarios. The resolution of the dataset used in this paper is 5 min which is to comply with the future LV network's operational requirements, as the observability of the LV network is increasing via the adoption of smart meters, making it essential to forecast the load variation up to minutes scale for low inertia network operation parameters such as voltages and frequency changing in the same time frame.
3. The proposed model is compared with several cutting-edge approaches, such as LSTM and hybrid CNN-LSTM. It has demonstrated a significant reduction in forecasting errors at different types of data and different DSO scenarios.

1.4. Paper organization

The rest of this paper is structured as follows: Methodology is presented in Section 2, Experimental setup is shown in Section 3, results and discussion is presented in Section 4, and finally, the conclusion in Section 5.

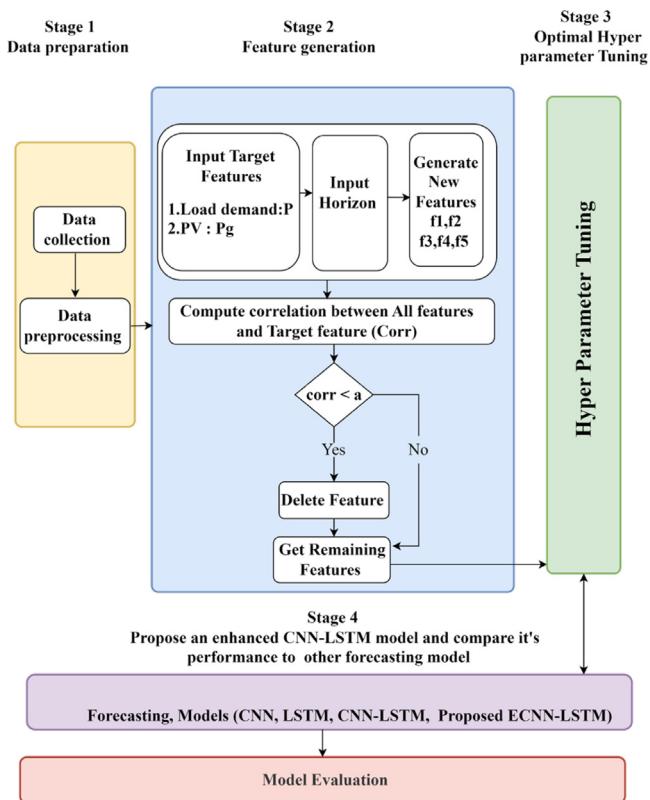


Fig. 1. Overall structure of the proposed method.

2. Methodology

The flowchart of the proposed method, consisting of four stages, is presented in Fig. 1. Firstly, data preparation consists of data collection and data preprocessing. In the second stage, a novel feature generation technique is developed to generate a new feature from the provided data, which is the input target feature (P and Pg) as illustrated in the next section. After that, all features, including the originally provided and newly generated features, are assessed using the Pearson correlation matrix as expressed in Eq. (8). Thus, the algorithm automatically keeps the most correlated features while eliminating the less correlated ones based on the chosen correction threshold factor "a" as shown in Fig. 1. FG and FS processes are further explained in Section 2.2. Moreover, in the third stage, a TPE algorithm is used for hyperparameter tuning to identify the optimal parameter values for each specific DL model to minimize the error and the computational time. In the last stage, an enhanced CNN-LSTM (ECNN-LSTM) DL forecasting model is developed. Furthermore, after setting up all the data, the developed feature generation technique is tested with six different forecasting techniques, including ARIMA, SVM, CNN, LSTM, CNN-LSTM, and the proposed ECNN-LSTM. Each stage is further explained in the following sections.

2.1. Stage 1 – Data collection and preprocessing

2.1.1. Data collection

2.1.1.1. Residential load demand data. The term '**load profile**' used in this paper refers to the load's time-varying active and reactive power during a given period (e.g., a day). These load profiles can be obtained from direct measurements (e.g., smart meter readings) if available or from time-of-use statistics of individual

appliances that represent the type of customer to be modeled. In this research, CREST demand model software (McKenna et al., 2015), a publicly available tool, is utilized to generate a pool of 100 load demand profiles based on the approach adopted in Richardson et al. (2010). This tool gets the parameters such as the number of occupants on the property, season, and type of day (weekend or weekday) as input to build daily load profiles for each appliance (from a set of appliances typically available for residential customers in the UK). In this 'component-based approach', the load profile of each appliance, $p_{app,i}^t$ is aggregated to produce a customer level time-varying load profile $P_{customer}^t$, as shown in Eq. (1) where a set of N appliances are considered. The same aggregation procedure is applied for the reactive component of the total customer load. In addition, the tool is used to generate weather data (e.g., temperature and solar irradiation) for each load profile at the chosen location in the UK. The location chosen to be in an urban area in England at a latitude of 52.8 and longitude of -1.2. The simulation was performed 31 times considering the week and weekend days to generate load profile data for 100 houses in July 2021. The resulting data was stored and rearranged to create a monthly load profile for each house.

$$P_{customer}^t = \sum p_{app,i}^t \quad \forall i \in N, t \in T \quad (1)$$

Therefore, the approach allows the production of pools of 100 load profiles for a month to be used as historical data that can then be used for L-STF proposed methodology assessment. The time resolution of the monthly generated load profile is one minute, so each profile (dataset) consists of 44 640 data points.

2.1.1.2. PV data. The dataset used to test the proposed model for the PV system was collected from Photovoltaic (2014). This dataset contains voltage, current, power, energy, and weather data from low-voltage substations and domestic premises with high uptake of PV-embedded generation. Data was collected as part of the project run by UK Power Networks, "Validation of Photovoltaic Connection Assessment Tool" (UK Power Networks, 2015). The project collected a rich dataset at domestic sites with Solar Panels. The dataset comprises 25,775 days of data and over 171 million individual measurements. This paper has extracted a specific set of PV data to test the proposed model.

2.1.2. Data description and preprocessing

2.1.2.1. Dataset description. The complete dataset used in this paper comprises two input datasets: load and PV. The residential load demand dataset consists of 100 profiles, each containing one month of data with a 1-min time resolution. The features of the load demand dataset are detailed in Table 2. All load data features in Table 2 were converted from 1-min to 5-min resolution to reduce the computational time needed for forecasting while maintaining the quality of the predictions.

The reason for converting the load dataset to a 5-min resolution is that this time window can capture all variations in load demand. Additionally, this conversion reduces the data points in each load profile from 44,640 to 8920, significantly reducing the computational time required for forecasting without compromising its quality. It is worth noting that most of the previous state-of-the-art approaches (Aprillia et al., 2020; Budin et al., 2022; Dai et al., 2022; du Plessis et al., 2021; Huang and Kuo, 2019; Jahangir et al., 2020; Kabilan et al., 2021; Kim and Cho, 2019; Korkmaz et al., 2021; Laouafi et al., 2017; Li et al., 2020; Mishra et al., 2020; Motepe et al., 2019; Razavi et al., 2020; Shi et al., 2017; Tan et al., 2022; Trivedi et al., 2022; Valgaev et al., 2017; VanDeventer et al., 2019; Yang et al., 2019; Zang et al., 2018) used a 1-h time resolution to perform L-STF and/or PV-STF due to the limited data resolution in those studies. However, the active network metering infrastructure provided by smart meters

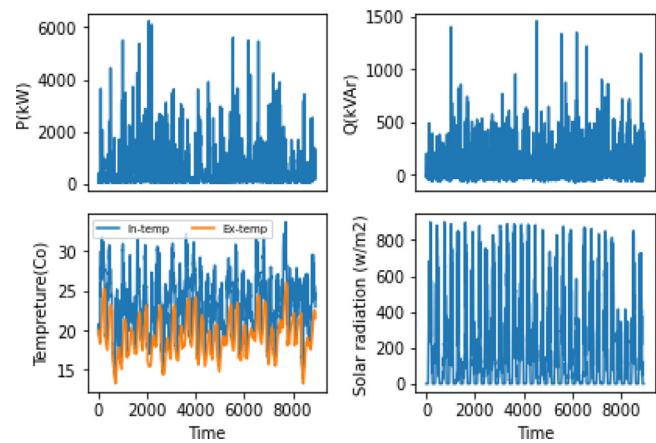


Fig. 2. Residential load demand data main features.

Table 2
Features of load dataset.

Feature	Unit	Abbreviation	Resolution
Date	-	Date	-
Time	min	Time	-
Active power	kW	P	1 min
Reactive power	kV Ar	Q	1 min
Solar radiation	W/m ²	Irr	1 min
Internal temperature	°C	In-temp	1 min
External temperature	°C	Ex-temp	1 min

Table 3
Features of the PV dataset.

Feature	Unit	Abbreviation	Resolution
Date	-	Date	-
Time	min	Time	-
Power generated	Kw	P-gen	1 min
Solar radiation	W/m ²	Irr	30 min
Internal temperature	°C	In-temp	30 min
External temperature	°C	Ex-temp	30 min
Wind speed	m/s	WS	30 min
Pressure	Bar	Bar	30 min

can record load profiles at a 5-min resolution. Recording load variation at this scale is crucial because other network operation parameters, such as voltages, can also change at a 5-min scale and are highly correlated with load variation. For example, an increase in load demand can lead to voltage reduction.

Fig. 2 Presents a graphical overview of the main features of a one-month (July 2021) residential load profile for House Number 20. The data comprises 8928 data points, with a resolution of 5 min.

Table 3 outlines the features included in the PV dataset, which can be divided into two categories. The first set of features has a resolution of 1 min, while the second category has a resolution of 30 min, as indicated in Table 3. Table 3 highlights that the PV weather data, such as solar radiation, temperature, wind speed, and pressure, need to be converted from a 30-min resolution to a 5-min resolution to fill in the missing data rows, as these features are only available in 30-min intervals. On the other hand, PV generation data was converted from 1-min to 5-min resolution.

Furthermore, Fig. 3 illustrates the main features of a one-month PV dataset, which comprises 8928 data points due to the data resolution being converted to 5 min. The figure illustrates the key characteristics of the dataset.

2.1.2.2. Training and testing data. The forecasting in this research aims to provide short-term predictions. The dataset consists of

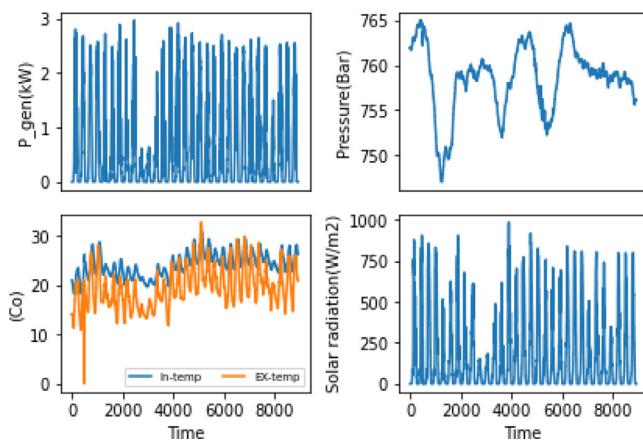


Fig. 3. PV data main features.

31 days of data, where each day has 288 data points, with a resolution of 5 min per data point. The data is split into 30 days for training, consisting of (8640) data points, and one day for testing, consisting of (288) data points. This approach is applied to all the forecasting exercises in this research, including the residential load demand, three-phase feeders load, three-phase substation load, and PV generation.

2.1.2.3. Data preprocessing. The preprocessing steps applied for both PV and load data are listed as follows:

- Merging the date and time to create a single date-time feature.
- Converting the data-time to index
- Converting every feature to a numeric value
- Normalization and scaling the data so that all the data are within the same ranges
- Sliding window

The first and the second steps of the preprocessing combine date and time features to generate the new feature to be the index.

In the third step, all the values in the dataset are converted to numeric values to be prepared for further forecasting. In the fourth step, a Min-Max normalization technique has been applied as the dataset features have different ranges, as shown in Figs. 2 and 3. Normalization is a standard preprocessing approach for bringing all feature values to a common scale to reduce large feature dominance and improve convergence. P, Q, and Radiation all have a maximum value of around 6000, while internal and external temperatures all have a maximum value of around 35, as shown in Fig. 2.

Therefore, all the data normalized in the range of 0–1 using the Min-Max normalization strategy, which linearly transforms X to X_{scaled} as expressed in Eq. (2) (Fekri et al., 2021).

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2)$$

where X is the original feature value, X_{min} and X_{max} are the minimum and maximum value of that feature, and X_{scaled} is the normalized value. In the last step, the sliding window technique (Kim and Cho, 2018) applied on all data inputs.

2.2. Stage 2–Feature generation and selection

The performance of the forecasting model depends on data quality and features used to train the model. At the training stage, the model learns from input data by mapping features

against a target. Too many or irrelevant features can make model learning processing difficult and complex, affecting overall forecasting accuracy. FS is a process of classifying a set of features most relevant to the target. FS improves forecasting accuracy, reduces training time, and removes data redundancy. However, some input features, such as sudden load demand spikes, might be complex and non-linear.

Therefore, a novel FG technique is developed in this section. A set of equations is designed to generate an additional feature from the existing historical dataset (i.e., P : load demand, P_{gen} : PV generation). The aim is to reduce the complexity of the features' relationship with the target feature. Hence, the newly generated features are designed to be highly correlated to the target feature, which reduces the overall nonlinearity and the complexity of the relationship between features. The proposed Eqs. (3)–(7) aim to:

(1) **Identify the variation pattern** of load demand and PV generation that is represented by the resolution difference feature(f_1) in Eq. (3).

(2) **Identifying the sudden change** in load demand and PV generation, either increment or decrement, via finding the maximum and minimum instance at a selected set of data points (1-h data, 12 data points in our case), and that is mathematically expressed in Eqs. (5)–(7)

(3) **Identify the general variation pattern** of load demand and PV generation by zooming out on the data by finding the hourly average (average of 12 data points in our case). Hourly average feature (f_2) will allow the DL model to have two phases of learning: (a) learn the data variation pattern (Big picture) through a smaller amount of data (743 data points instead of 8920 data points each month) and then (b) learn the detailed variation in data via the real dataset points (8920 data points). This is mathematically expressed in Eq. (4). The proposed Eqs. (3)–(7) These are further explained in the subsections below.

2.2.1. Resolution difference (f_1)

This newly generated feature is the difference between the current target value and the previous target value. This is done based on the assumption that the current target value at a particular time is highly correlated with the difference between the current and the previous target values. This feature will help identify the load growth or reduction pattern over the day, improving the learning ability of DL models. The process can mathematically be expressed in Eq. (3).

$$f_{1i} = tf_i - tf_{i-1} \quad (3)$$

where tf is the target feature and i is the index of the datapoints in the dataset.

2.2.2. Hourly average (f_2)

This feature assumes that the target feature at a particular time will be correlated with the average target value in that hour.

This can be mathematically expressed in (4).

$$f_{2i} = \frac{\sum_{j=i}^{n=12} tf_j}{12} \quad (4)$$

As the time resolution is 5-min, each hour has 12 data points. Therefore, in (4) out denominator is set to be 12.

2.2.3. Maximum hourly value (f_3)

This feature assumes that the target feature (P , P_{gen}) at a particular time will correlate with the maximum target value in that hour. This can be mathematically expressed in (5)

$$f_{3i} = \max(\{tf_i, \dots, tf_n\}) \quad (5)$$

where, n is the number of data point at every hour, which is 12 in our case.

2.2.4. Minimum hourly value (f_4)

This feature assumes that the target feature (P , P_{gen}) at a particular time will be correlated with the main target value in that hour. This can be mathematically expressed in (6)

$$f_{4i} = \min(\{tf_i, \dots, tf_n\}) \quad (6)$$

where, n is the number of data point at every hour, which is 12 in our case.

2.2.5. Mid-range hourly value(f_5)

This feature assumes that the target feature (P , P_{gen}) at a particular time will be correlated with the average of *max* and *min* target values in that hour. This can be mathematically expressed in (7)

$$f_{5i} = (\min(\{tf_i, \dots, tf_n\}) + \max(\{tf_i, \dots, tf_n\}))/2 \quad (7)$$

After adding the newly generated features, Pearson correlation is applied for feature selection (FS) by measuring the linear relationship between all features, including the newly generated features. On a scale of $+1$ to -1 , $+1$ represents a positive relationship between two variables, -1 represents a negative relationship, and 0 indicates no relationship. The Pearson Correlation Coefficient (corr) between two variables, x , and y , is represented in Eq. (8) (Elahe et al., 2022). The algorithm uses a correlation threshold factor “ a ” to decide which features to keep and which to eliminate. In this study, we have chosen a threshold of $a = 0.2$, which means that a coefficient of 0.2 or higher indicates a potential relationship between the target feature and other features. For example, if the correlation factor between a specific feature (f_1) and the target feature is lower than 0.2 , the algorithm will consider that feature uncorrelated and remove it. On the other hand, if the correlation factor is higher than 0.2 , the algorithm will keep that feature. This process is visualized in Fig. 1.

$$\text{Corr}(x, y) = \frac{\sum_{i \in I_x \cap I_y} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i \in I_x \cap I_y} (x_i - \bar{x})^2} \times \sqrt{\sum_{i \in I_x \cap I_y} (y_i - \bar{y})^2}} \quad (8)$$

where, x_i y_i are the x and y variables sample, respectively; \bar{x} and \bar{y} are the mean of values in x and y variables, respectively.

For instance, Figs. 4 and 5 present the correlation matrix for both the load profile data of house number 20 and PV data, which include the already given features and the newly generated features (f_1-f_5).

As shown in Fig. 4, the target feature (P) is the fourth feature from the left, and it can be observed that the correlation factor for all features, except for internal temperature (In-temp) and irradiation (Irr), is higher than 0.2 . In such a scenario, the algorithm will eliminate the In-temp and Irr features and retain the other eight features, which include the target feature (P), external temperature, and reactive power (Q), as well as the newly generated features (f_1-f_5). It is important to note that since we have 100 load data profiles, each profile has its correlation matrix. The process of selecting the correlated features occurs separately for each profile, and it is expected that not all load profiles will have the same features. These features are chosen automatically based on their correlation with the target feature.

2.3. Stage 3–Hyperparameter tuning

The process of selecting the best set of hyperparameters for the machine learning algorithm is known as hyperparameter tuning. This paper uses the TPE algorithm to find the optimal hyperparameter to avoid the time-consuming task of manually tuning the hyperparameters. The TPE algorithm is based on Sequential Model-Based Global Optimization, which finds the best solution for a convex optimization problem and saves the scores

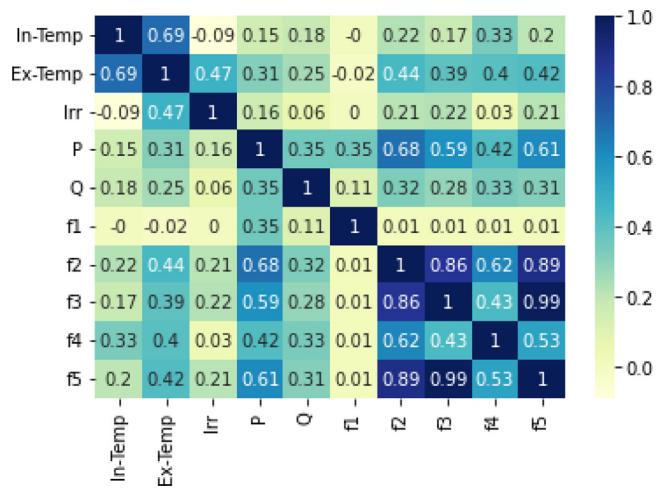


Fig. 4. Load data person correlation matrix.

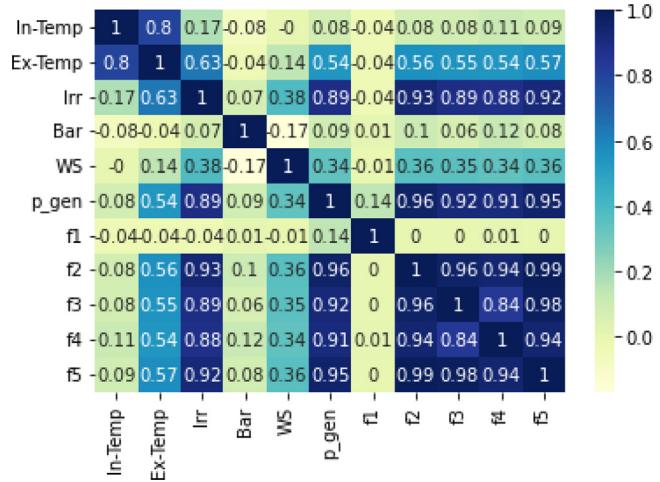


Fig. 5. PV data person correlation matrix.

for each configuration in the database as a set of tuples via an iterative process (score, configuration). The user initially defines the search space with a set of parameters to be tuned; mathematically, it is determined by a continuous and convex function. A loss function is evaluated for each configuration setting using the number of observations. The search space with the highest score is extracted, and the new space is redefined for further sampling. This procedure is repeated until the highest overall score, or early stopping, is achieved. The detailed algorithm is explained in Bergstra et al. (2011, 2013). In this research, the TPE algorithm was accessed through the OPTUNA Python library (Akiba et al., 2019) to model and determine the optimal hyperparameters.

2.4. Stage 4–Enhanced CNN-LSTM

This section describes the proposed short-term forecasting method for PV generation and load demands. The inputs are multivariate for all data using the features provided in Tables 2 and 3 and the newly generated features (f_1-f_5).

The structure of the ECNN-LSTM model is shown in Fig. 6. As illustrated in Fig. 1, if the proposed FG technique is not used, a constant number of six features is applied, as given in Tables 2 and 3. When the proposed FG technique is applied, a different set of features are automatically selected for each forecasting exercise. The enhanced CNN-LSTM method consists of

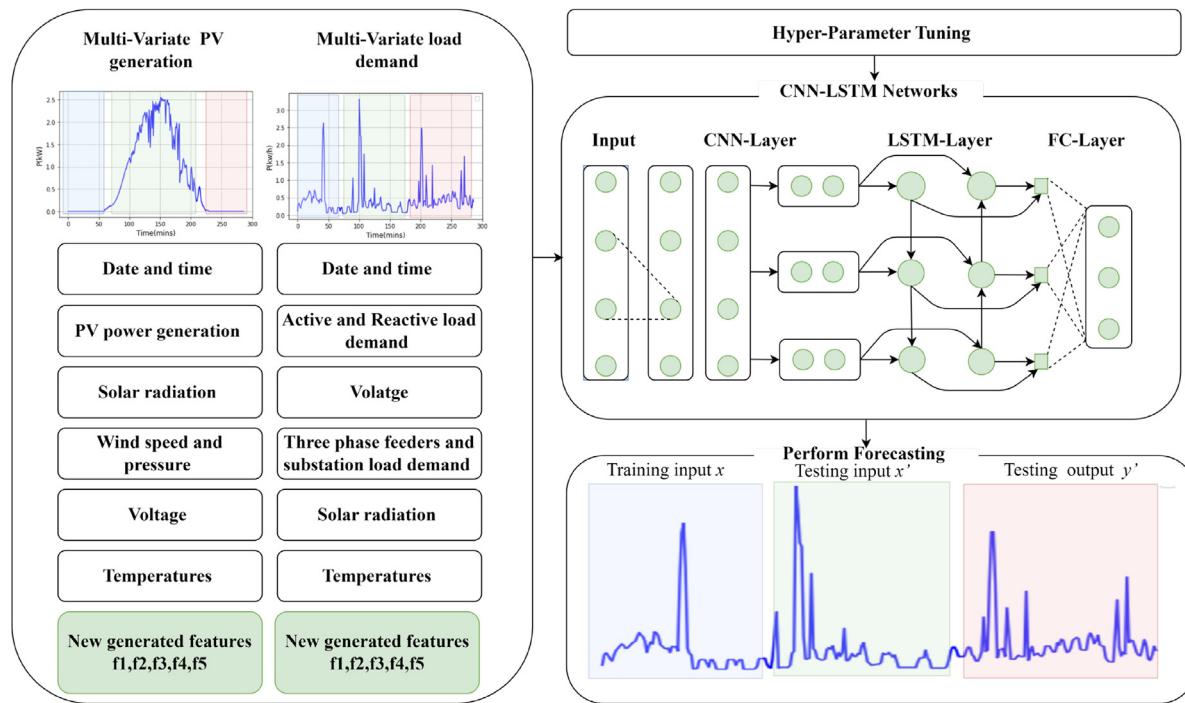


Fig. 6. The structure of the proposed enhanced CNN-LSTM model.

three layers, the convolutional layer, the LSTM layer, and the fully connected layer. The ECNN-LSTM model learns from input data that has been preprocessed by the sliding window algorithm every five minutes (Kim and Cho, 2018). The spatial characteristics of multivariate time series variables are extracted from the CNN convolution layer and then passed on to the LSTM layer after extracting the representative features from the input data. The LSTM layer models the irregular time information using the transmitted spatial features. Finally, the output of the LSTM layer is fed into two fully connected (FC) layers with a Rectified Linear Activation Function (ReLU) layer. In the FC layer, the enhanced CNN-LSTM method can predict PV and load demand profiles. All these layers are further explained in the next subsections. It is worth noting that the ECNN-LSTM method differs from the regular CNN-LSTM architecture proposed in Kim and Cho (2019) in that the pooling layers were not used. This is because the pooling layer reduces the input data's dimensions for the next layer. However, applying a max pooling layer to the input data will extract only the highest value from the pooling region and discard smaller values, potentially causing the loss of sensitive information (Su et al., 2015; Zeiler and Fergus, 2014). Moreover, the average pooling layer computes the average value for the selected pooling region. Hence the representative features will be manipulated after computing the average value (Bera and Shrivastava, 2020; Nirthika et al., 2022). Another key difference with the regular CNN-LSTM is that the proposed model can correctly adjust itself when the number of features changes.

2.4.1. Convolutional Neural Network (CNN) layer

The convolutional layer is the core block of the CNN model. It is used to extract characteristic features from data. The convolutional layer employs shared weights and the local receptive field methods via the convolution operation (Han et al., 2019). The input data is convolved with one or more convolutional filters in the convolutional layer, yielding an output feature map for each filter (Alqatawneh et al., 2021). The local receptive field size and shared weights across all convolutional filters in a specific convolution layer are the same. The output of the convolutional

layer can be expressed mathematically as in Eq. (9) (Han et al., 2019):

$$x_j^l = f \left(\sum_{i=1}^M x_i^{l-1} * w_{ij}^l + b_j^l \right) \quad (9)$$

where x_j^l is the output value of j th neuron in the feature map at the l th layer, f is the activation function, $*$ denotes the convolution operation, x_i^{l-1} is the input feature map at $(l - 1)$, w_{ij}^l is the convolutional filter connected the i th input feature map at $l - 1$ with j th feature map at l . Also, b_j^l refer to the bias.

2.4.2. Long Short-Term Memory network (LSTM) layer

Long Short-Term memory (LSTM) network is a type of Recurrent neural network (RNN) developed by Hochreiter (Kim et al., 2020) to overcome the exploding and vanishing gradient problems in RNN. It is a special Neural Network with the ability of learning long-term dependency. Each LSTM has three main gates that serve as input, output, and forget gates in each LSTM network. Deep LSTM networks have a great potential to learn the pattern of complex phenomena with high stochastic behavior in this regard. The basic architecture of LSTM is described below.

As shown in Fig. 7, the structure of an LSTM unit consists of a cell and three gates; the cell works as the LSTM unit's memory; gates are used to manage the information flow within the LSTM unit, including a forget gate f_t , input gate i_t and output gate o_t .

The current input data vector x_t , the previous hidden state h_{t-i} , and the prior cell state c_{t-1} are all sent to each gate in the LSTM unit (Kim and Cho, 2019).

As shown in Fig. 7, the Forget gate f_t determines the internal information needs to be removed from the previous cell state c_{t-1} , and it can be calculated as follows (10) (Kim and Cho, 2019; Kim et al., 2020):

$$f_t = \text{sigmoid}(w_{xf} x_t + w_{hf} h_{t-i} + b_f) \quad (10)$$

Input gate i_t decides whether new information is stored in the current cell state c_t , and its can be calculated as follows Kim and Cho (2019) and Kim et al. (2020):

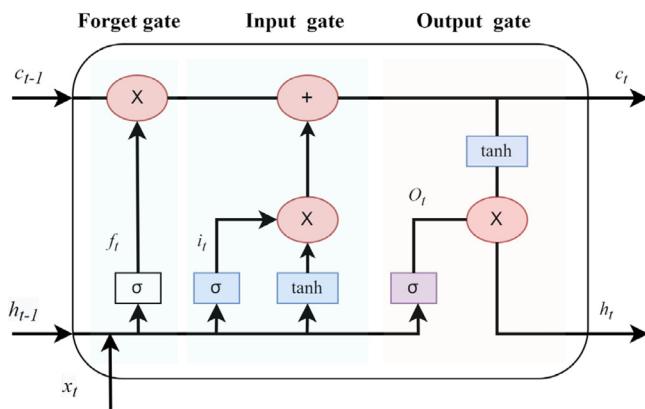


Fig. 7. LSTM structure.

The input gate i_t decides whether new information is stored in the current cell state c_t and can be calculated as follows in (11)–(12).

$$i_t = \text{sigmoid}(w_{xi}x_t + w_{hi}h_{t-1} + b_i) \quad (11)$$

$$\bar{c}_t = \tanh(w_{xc}x_t + w_{hc}h_{t-1} + b_c) \quad (12)$$

Output gate determines which information will be passed through the gate and get into the rest of the network, and it can be calculated as follows in (13)–(15) (Kim et al., 2020):

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \bar{c}_t \quad (13)$$

$$o_t = \text{sigmoid}(w_{xo}x_t + w_{ho}h_{t-1} + b_o) \quad (14)$$

$$h_t = o_t \otimes \tanh(c_t) \quad (15)$$

2.4.3. Rectified linear activation function (relu)

An activation function layer is applied to the output feature map of the LSTM layer. It enables the network to perceive non-linear data expressions, allowing it to handle various complex problems (Alqatawnah et al., 2020). The ReLU function was created to overcome the vanishing gradient problem in *Tanh* and *sigmoid* functions; it is used in this design and can be mathematically expressed in (16) (Zhang et al., 2017).

$$f(y_i) = \max(0, y) = f(x) = \begin{cases} y, & y > 0 \\ 0, & y \leq 0 \end{cases} \quad (16)$$

where $f(y_i)$ is the output of the ReLU function.

2.4.4. Fully connected layer (FC)

This layer has all its inputs connected to every activation unit in the layer above them. The final few layers in most well-linked machine learning models are fully connected layers that combine the data gathered by earlier layers to create the output. It takes the second-longest time after the Convolution Layer. Mathematically, it can be described in (17).

$$f(x) = xA^t + b \quad (17)$$

2.4.5. Model architecture

The architecture of the proposed enhanced CNN-LSTM model is provided in this section. CNN and LSTM are the key layers of architecture. The first layer is the CNN layer which is one-dimensional (Conv1D). Conv1d has an input layer that allows the model to receive the input variables. For L-STF, there can be a different number of input variables due to the FG and FS preprocessing techniques, as discussed in Section 2. For PV-STF, the input variables are described in Table 3. The convolutional layer produces an output that is passed through the LSTM layer.

Table 4
The proposed method architecture.

Layer	Kernel size	Filters	Stride	# Params
Conv1d	1	64	1	3
LSTM	–	64	–	166,400
Linear (200)	–	–	–	40,200
Activation (ReLU)	–	–	–	–
Linear (200)	–	–	–	201
Total Parameters	–	–	–	206, 804

All convolutional layers have kernels and strides. The kernel is a matrix that iterates over the input data, conducts a dot product operation with a subregion of the input data, and outputs the result as a matrix of dot products. Stride is the number of pixels that shifts over the input data. We used 1 for both the number of kernels and strides. Table 4 shows the full description of the proposed architecture.

2.5. Other forecasting models

2.5.1. Support vector regression (SVR)

An SVR model is designed to solve regression problems with multiple predictors $X = \{x_i\}_{i=1}^n$ where n is the number of predictor variables and each x_i has n variables. These are linked to an objective variable $y = \{y_i\}_{i=1}^N$. The fundamental concept of SVR is to transform the input data into higher dimensional feature space using a non-linear mapping function ϕ . Consequently, the matrix X is converted to higher-dimensional feature space, in accordance with the original (Al-Musaylh et al., 2018).

With an SVR model, a nonlinear regression problem can be expressed as in Eq. (18) (Awad and Khanna, 2015; Wang et al., 2009)

$$y = f(X) = \omega\phi(X) + b \quad (18)$$

Here b is a constant, ω is the weighted vector, and $\phi(x_i)$ represent the mapping function employed in the feature space. The coefficient b and ω are estimated through minimization process outlined in Eq. (19) (Al-Musaylh et al., 2018):

$$\text{Minimize } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{i=n} (\xi_i + \xi_i^*) \quad (19)$$

Subjected to

$$y_i - \omega^T \phi(x_i) - b \leq \varepsilon + \xi_i \quad (20)$$

$$\omega^T \phi(x_i) + b - y_i \leq \varepsilon + \xi_i^* \quad (21)$$

$$\xi_i, \xi_i^* \geq 0 \quad (22)$$

where C and ε are predetermined model parameters. The term $\frac{1}{2} \|w\|^2$ represent the smoothness of the function and constant C influence the trade-off between an approximate error and the weight vector norm $\|\omega\|$. The slack variables ξ_i and ξ_i^* account for the deviation between the predicted and actual values, with ε representing the maximum allowed deviation. Penalizing occurs for samples whose prediction deviate by at ε from their true target. The objective function is penalized by ξ_i or ξ_i^* , depending on whether their predictions lie above or below the ε tube (Awad and Khanna, 2015).

By applying Lagrangian multipliers and optimizing conditions, we obtain the non-linear regression function shown in Eq. (23) (Al-Musaylh et al., 2018):

$$f(X) = \sum_i^n (\alpha_i - \alpha_i^*) K(x_i, x_j) + b \quad (23)$$

where α_i and α_i^* are Lagrangian multiplier and they can be found by solving the dual form of Eq. (19). $K(x, x_i)$ are the Kernel

function (any function that satisfied the Mercer's theorem can be used) (Wang et al., 2009). In the form of $K(x_i, x_j) = \phi(x_i)\phi(x_j)$, the Kernel function perform the non-linear mapping, implicitly projection the training vector into a higher dimensional space using the ϕ function. Both $x_i, x_j \in X$. The data point associated with the decision surface, commonly known as "support vector" are the ones laying closest to it (Al-Musaylh et al., 2018).

2.5.2. Autoregressive Integrated Moving Average (ARIMA)

The Autoregressive Integrated Moving Average (ARIMA) model is a widely used approach for time-series forecasting, which relies on past data to predict future values. The ARIMA model combines two types of linear regressions: Autoregressive (AR) and Moving Average (MA) (Al-Musaylh et al., 2018). The AR model is represented as follows:

$$y_t = c + a_1 y_{t-1} + \dots + a_p y_{t-p} + u_t \quad (24)$$

In Eq. (24) a_1, \dots, a_p are the autoregressive parameters, c is a constant term, p is the order of the autoregressive process, y_t is the current value being predicted, and u_t is the white noise or error term (Yin et al., 2022).

Similarly, the Moving Average (MA) model can be written as:

$$y_t = \mu + u_t + m_1 u_{t-1} + \dots + m_q u_{t-q} \quad (25)$$

Here, m_1, \dots, m_q are the moving average parameters, q is the order of the moving average process, u_t, \dots, u_{t-q} are the white noise or error terms, and μ is the expectation of y_t .

By combining these two models with the same training data, the ARIMA model, denoted as ARIMA (p, q), is obtained:

$$y_t = c + a_1 y_{t-1} + \dots + a_p y_{t-p} + u_t + m_1 u_{t-1} + \dots + m_q u_{t-q} \quad (26)$$

In Eq. (26), p represents the autoregressive terms, q represents the moving average terms, and y_t is the predicted value. The ARIMA model assumes that the time-series data exhibits statistical stationarity, meaning that statistical properties like mean, variance, and autocorrelation remain constant over time. However, if the training data shows non-stationarity, which is often the case with real-life data, such as predictor signals, the ARIMA model requires differenced data to transform it into a stationary series. This is indicated by ARIMA(p, d, q), where d represents the degree of differencing (Yin et al., 2022).

3. Experimental setup

Several forecasting architectures (ARIMA, SVM, CNN, LSTM and CNN-LSTM) are compared in this study to evaluate the effectiveness of the proposed enhanced CNN-LSTM(ECNN-LSTM) architecture. Also, several cases have been investigated to evaluate the effectiveness of the proposed ECNN-LSTM compared to the existing ones in the literature. These comparisons are as follows:

- Analysis of the state of art methods and the proposed enhanced CNN-LSTM with and without the proposed FG technique on the residential load data
- Three-phase feeder and substation load forecasting analysis with and without the proposed FG technique
- Analysis of the PV data with and without the proposed FG technique

3.1. Hardware and software requirements

All the experiments in this study were conducted on the core i7 computer system with 16 GB RAM. Pytorch python library (Paszke et al., 2019) is used to develop both the state of art methods and the proposed method.

Table 5

Optimal hyper parameter for residential load demand (House no. 20).

Hyper parameter	Learning rate	Epochs	Batch size
Without FG	0.002809444815538457	5	76
With FG	0.009969079572828235	4	70

3.2. Evaluation metrics

The following are the evaluation metrics used to validate the proposed ECNN-LSTM model.

$$MAE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n} \quad (27)$$

$$MSE = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n} \quad (28)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \quad (29)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{(y_i - \hat{y}_i)}{y_i} \right| \quad (30)$$

where y_i is the actual value, \hat{y}_i is the predicted value, n , is the total number of the predicted points. Mean absolute error (MAE) is popular because the error value units match the predicted target units. Unlike RMSE, the changes in MAE are linear. Thus, in MAE error scores increase linearly with the increase in errors. The mean squared error (MSE) and the root mean squared error (RMSE) is the alteration between the original and predictable values. It is mined by forming the mean formed error of the dataset. The mean absolute percent error (MAPE) is the accuracy of a prediction as it measures the size of the error.

3.3. Hyper-parameter tuning

Several parameters of the proposed ECNN-LSTM architecture were optimized using the TPE algorithm, and the one with the best forecasting measure was selected as the optimal parameter of the final model. This process is applied for each forecasting exercise accordingly. The obtained hyper parameter for each forecasting exercise is provided below:

3.3.1. Residential load demand forecasting hyper parameter

The hyperparameter tuning for load profile 20, which is included in the presented results, is detailed in Table 5 below. However, it is important to note that each household forecasting scenario has its own unique optimal set of hyperparameters due to variations in demand profile data.

3.3.2. Feeders forecasting hyper parameter tuning

The optimal hyperparameters for feeder three three-phase load profile are displayed in Table 6. It is important to highlight that each phase possesses its own set of optimal parameters, and the selection of optimal parameters also relies on the type of features used. Therefore, the results are presented with and without FG to account for this factor.

3.3.3. Substation forecasting optimal hyperparameter

Table 7 presents the optimal hyperparameters for three-phase substation forecasting, including results with and without FG.

3.3.4. PV generation forecasting optimal hyperparameter

Table 8 presents the optimal hyperparameters obtained for the PV generation forecasting exercise.

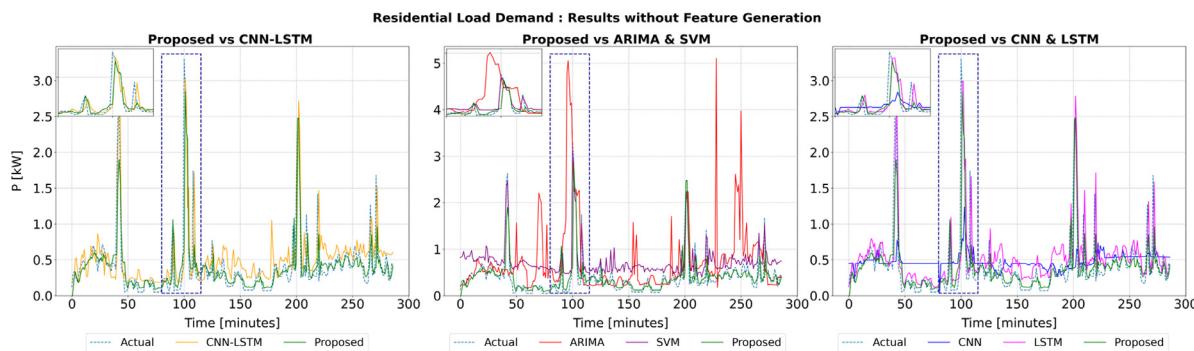


Fig. 8. . Real time residential load forecasting without FG at house No. 20.

Table 6
Optimal hyper parameter for the three-phase feeder.

FG status		Learning rate	Epochs	Batch size
Without FG	Phase A	0.002290706465838941	8	63
	Phase B	0.007399592922255475	8	69
	Phase C	0.0038155849522862914	8	34
With FG	Phase A	0.0005026313034767111	7	24
	Phase B	0.005592589506409543	9	251
	Phase C	0.0056303807933502405	7	148

Table 7
Optimal hyper parameter for the three-phase substation.

FG status		Learning rate	Epochs	Batch size
Without FG	Phase A	0.002291229253830006	16	62
	Phase B	0.005418158938896748	4	44
	Phase C	0.0005846726407128307	13	17
With FG	Phase A	0.00910265559351474	15	154
	Phase B	0.005885119666293963	7	42
	Phase C	0.006090860741317217	11	44

Table 8
PV generation optimal hyper parameter.

Hyper parameter	Learning rate	Epochs	Batch size
Without FG	0.007598302974106191	14	88
With FG	0.0004902927091997263	20	38

4. Results and discussion

4.1. Residential load demand forecasting analysis

4.1.1. Results without FG

In this case, the results of the 100 and 400 residential load demand forecasting are presented in Tables 9 and 10, respectively. These tables provide average performance metrics for the 100 and 400 residential load profiles, facilitating a comparison of six models: ARIMA, SVM, LSTM, CNN, CNN-LSTM, and the proposed ECNN-LSTM. Notably, the ECNN-LSTM model demonstrates the smallest average MAE, MSE, RMSE, and MAPE values compared to all other methods in the 100 and 400 load demand forecasting scenarios. However, it is worth noting that Tables 9 and 10 also indicate that the ARIMA, SVM, and CNN models exhibit the highest average error values when compared to other methods for the 100 and 400 residential load demand forecasting, respectively.

For instance, an examination of Table 9 confirms that in the case of 100 load demand forecasting, the proposed method exhibits the best forecasting performance, yielding significant improvements over the CNN-LSTM model. Specifically, it demonstrates a 26% reduction in MAE, a 19% reduction in MSE, a 10% reduction in RMSE, and a 48% reduction in MAPE. Moreover, it also shows improvements over the LSTM model with a 20%

decrease in MAE, a 3% decrease in MSE, a 2% decrease in RMSE, and a 1% decrease in MAPE.

In the case of 400 load demand forecasting, Table 10 confirms that the average MAE, MSE, RMSE, and MAPE experience slight increases compared to the values reported in Table 9. Notably, the ECNN-LSTM model, in particular, shows a 4% increase in MAE, a 39% increase in MSE, a 17% increase in RMSE, and a 22% increase in MAPE compared to the corresponding results in Table 9. These findings suggest a slight degradation in performance as the forecasting scenario transitions from 100 to 400 households, as demonstrated by the results in Table 10.

In addition, Tables 9 and 10 compare household prediction and training times for all the methods in the 100 and 400 household demand forecasting scenarios. It is evident that the total prediction and training times for all the methods are generally acceptable for both the 100 and 400 households, except for the CNN-LSTM model, which requires approximately 50 min for 100 households and 3.34 h for 400 households. On the other hand, the proposed ECNN-LSTM model demonstrates significantly faster performance, completing training and predictions in just 4 min for 100 households and 16 min for 400 households. Furthermore, the tables also illustrate that the SVM model has the shortest training and prediction times overall, followed by the CNN model. This highlights the efficiency of these models in terms of computation time.

Fig. 8 presents the forecasted results of the proposed method alongside other forecasting models, including CNN-LSTM, ARIMA, SVM, CNN, and LSTM. The data spans 300 points, representing 24-h intervals, for House No. 20. In comparing the proposed model with CNN-LSTM, it is important to note that both models capture the general pattern of the residential load profile. However, the proposed ECNN-LSTM demonstrates superiority by providing more accurate predictions for both peak and average consumption. The CNN-LSTM (Kim and Cho, 2019) exhibits lower accuracy due to the CNN pooling layer, which tends to discard sensitive information during the learning stage to reduce computational running time, resulting in less accurate predictions. Additionally, as mentioned in He and Sun (2015), the network's ability to detect hidden features from raw data improves with network depth. However, excessive layering can lead to a decrease in forecasting accuracy. Tables 9, 10, and Fig. 8 illustrate that using the CNN-LSTM architecture provided in Kim and Cho (2019) yields lower forecasting accuracy compared to ECNN-LSTM. The findings from Tables 9, 10, and Fig. 8 confirm that increasing the number of hidden layers in the CNN-LSTM (Kim and Cho, 2019) architecture adds more parameters to train and raises the risk of overfitting, ultimately resulting in less accurate forecasting compared to the proposed enhanced CNN-LSTM.

Furthermore, the figure clearly demonstrates the significant superiority of the proposed method over both the ARIMA and SVM models. It is evident that SVM and ARIMA exhibit the highest

Table 9
One hundred residential load forecasting results without FG.

Method	MAE (kW)	MSE (kW)	RMSE (kW)	MAPE (%)	Total 100 house prediction time (s)	Total 100 house training time (s)
ARIMA	0.4745	0.8095	0.8997	0.5665	2.1056	286.4597
SVM	0.3447	0.1469	0.3832	0.5129	0.8912	5.1889
CNN	0.2610	0.1826	0.4273	0.6870	2.5814	125.3990
LSTM	0.2254	0.1357	0.3683	0.4531	4.6209	775.3496
CNN-LSTM	0.2455	0.1624	0.4030	0.8621	10.7877	3018.8464
Proposed	0.1806	0.1312	0.3622	0.4481	3.0294	231.9624

Table 10
Four hundred residential load forecasting results without FG.

Method	MAE (kW)	MSE (kW)	RMSE (kW)	MAPE (%)	Total 400 house prediction time (s)	Total 400 house Training time (s)
ARIMA	0.4344	0.7165	0.8464	1.2278	8.6250	1187.2120
SVM	0.4447	0.2469	0.4968	0.6129	3.5620	21.6760
CNN	0.3350	0.3588	0.5990	0.8650	10.1890	490.4480
LSTM	0.2483	0.2340	0.4838	1.2531	18.2150	3058.5080
CNN-LSTM	0.2455	0.1624	0.4030	0.8621	43.1290	11991.7750
Proposed	0.1880	0.1826	0.4273	0.5485	12.0080	930.8190

Table 11
One hundred residential load forecasting results with FG.

Method	MAE (kW)	MSE (kW)	RMSE (kW)	MAPE (%)	Total 100 house prediction time (s)	Total 100 house Training time (s)
ARIMA	0.5086	0.8501	0.9220	1.4752	2.397487	758.0398
SVM	0.4757	0.3039	0.5512	0.6059	1.206682	9.798273
CNN	0.2249	0.1352	0.3677	3.4585	2.642493	101.7949
LSTM	0.1945	0.1056	0.3250	0.4050	4.76378	630.1021
CNN-LSTM	0.1992	0.1006	0.3171	0.4423	11.26056	2330.835
Proposed	0.1456	0.0821	0.2864	0.4302	3.149558	191.0317

Table 12
Four hundred residential load forecasting results with FG.

Method	MAE (kW)	MSE (kW)	RMSE (kW)	MAPE (%)	Total 100 house prediction time (s)	Total 100 house Training time (s)
ARIMA	0.4483	0.7408	0.8607	2.1974	9.1877	3171.0974
SVM	0.5211	0.4234	0.6507	0.7692	4.8721	41.1011
CNN	0.3027	0.3269	0.5718	0.9284	10.3718	403.6000
LSTM	0.1952	0.1488	0.3858	0.9213	18.5051	2498.6551
CNN-LSTM	0.2026	0.1718	0.4145	0.9901	44.2454	9328.9324
Proposed	0.1713	0.1596	0.3995	0.5115	12.3316	762.5839

errors, which aligns with the results reported in [Tables 9](#) and [10](#). Additionally, when comparing the proposed hybrid model with the individual CNN and LSTM models, it is noteworthy that while the proposed model outperforms them, both CNN and LSTM still achieve a satisfactory level of accuracy. However, it is important to highlight that the CNN model is less accurate compared to both the proposed model and LSTM.

4.1.2. Results with FG

[Tables 11](#) and [12](#) respectively present the results of 100 and 400 residential load demand forecasting with the implementation of the proposed FG technique across all methods. Notably, the FG technique has enhanced the accuracy of most methods in both 100 and 400 load demand forecasting scenarios, compared to the results presented in [Tables 9](#) and [10](#).

For instance, in the case of 100 household forecasting, applying the FG technique leads to improvements of 19% in MAE, 38% in MSE, 21% in RMSE, and 48% in MAPE for the CNN-LSTM method, as confirmed by the values in [Table 11](#). Similarly, the proposed ECNN-LSTM method demonstrates improvements of 19% in MAE, 37% in MSE, 21% in RMSE, and 4% in MAPE compared to the results without FG. On the other hand, it is worth noting that the accuracy of the ARIMA and SVM models is reduced when the FG technique is applied, compared to the results reported in

[Tables 9](#) and [10](#). This observation suggests that these two models are highly sensitive to increased data volume.

It is important to acknowledge that conventional ML models, as mentioned in [Kim and Cho \(2019\)](#), often require prior knowledge of the data and are prone to overfitting as the data volume increases. This aligns with the performance results of the SVM model in this case.

[Tables 11](#) and [12](#) also compare the training and prediction times for all models. It is evident that applying the FG technique has helped reduce the training time for all deep learning models, including CNN, LSTM, CNN-LSTM, and the proposed model.

For example, according to the results reported in [Tables 9](#) and [10](#), the proposed model required 4 min for 100 households and 16 min for 400 households. However, with the implementation of FG, the training time significantly decreased to 3.2 min for 100 households and 13 min for 400 households. This represents a reduction of 20% in computational time for both the 100 and 400 households forecasting cases.

[Fig. 9](#) presents the real-time forecasted results of the proposed method alongside other forecasting models, namely CNN-LSTM, ARIMA, SVM, CNN, and LSTM, utilizing the FG technique for house No. 20. It is evident that most deep learning models, including the proposed ECNN-LSTM, CNN-LSTM, CNN, and LSTM, have demonstrated improved accuracy compared to the results presented in [Fig. 8](#). Particularly, notable improvement can be observed in the CNN-LSTM model.

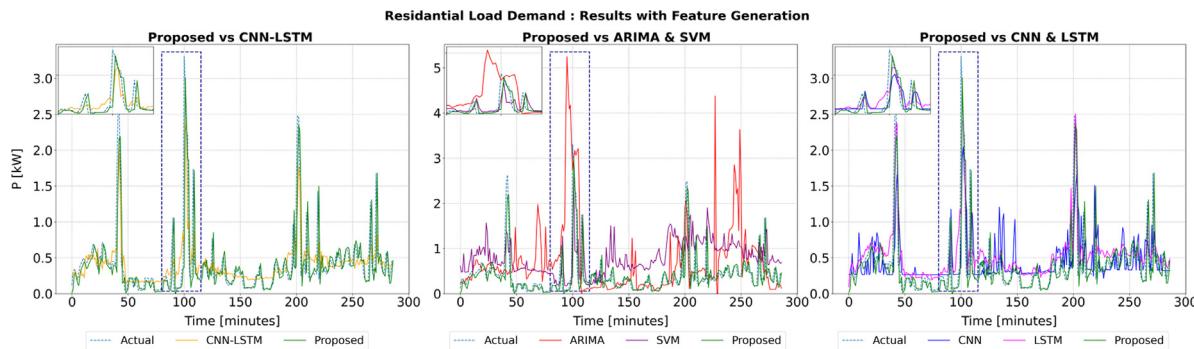


Fig. 9. Real time residential load forecasting with FG at house No. 20.

However, the application of the FG technique on the SVM and ARIMA models does not contribute significantly to their accuracy improvement. These models tend to be sensitive to the increased dataset input, and their accuracy remains comparatively low, which aligns with the results reported in the tables.

Furthermore, the CNN-LSTM model exhibits enhancement in forecasting both the peak value instance and the average consumption. Nevertheless, the proposed method outperforms all other approaches in terms of accuracy when forecasting the peak instance.

4.2. Three-Phase load demand forecasting analysis

In this section, the 100 load demand data generated is randomly allocated on a real three-phase LV network which consists of three feeders, as shown in Fig. 10. All feeders are three-phase and serve single-phase houses (represented by the black dots). The number of customers connected at each feeder is shown in Table 13. For the sake of simplicity, we assumed an equal distribution of customers across the three phases at each feeder. For example, feeder 1 has 157 customers, and since the 100 load demand profiles were randomly allocated, some customers were assumed to have the same load profile. Therefore, the customers were divided evenly among the three phases, with 52 customers assigned to phases A and B, and 53 to phase C.

Once the load demand profiles were allocated to each phase of the feeder, we aggregated the load demand for each profile at each data point (time) for phase A and performed the same process for the other feeder phases. This resulted in a new aggregated load profile for each phase at feeder 1. We applied the same process to feeders 2 and 3, generating a new load profile for each phase on each feeder, resulting in 9 profiles for all the feeders.

A similar approach was applied to generate the substation load profile for each phase. We aggregated the load demand for phase A from each of the three feeders and repeated the process for phases B and C. This process produced a new profile for each phase at the substation.

Since the original load profiles represent a time series for an entire month, the newly generated aggregated profiles for the substation and feeders maintain the same characteristics. These newly generated profiles were used to test the model for forecasting three-phase and substation load demand. This experiment aims to show the ability and superiority of the proposed ECNN-LSTM method in forecasting a three-phase load demand at both feeder and substation levels.

The proposed framework is designed to be tested on various DSO scenarios, including three-phase substation and feeder-level forecasting. Fig. 11 illustrates the differences in load demand data between single residential loads, feeder loads (aggregated loads of 90 customers at feeder 3), and substation loads (aggregated loads for 330 customers). The graph shows that the volatility of

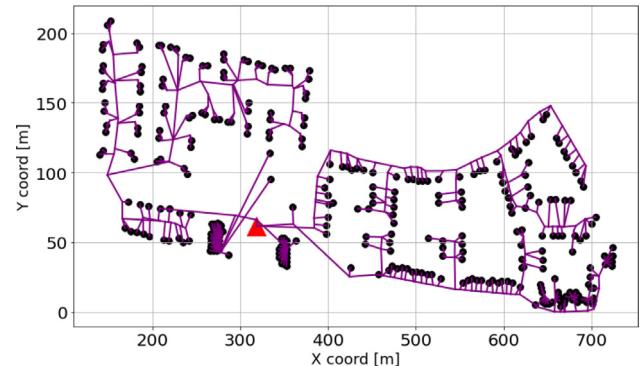


Fig. 10. Real three-phase LV network.

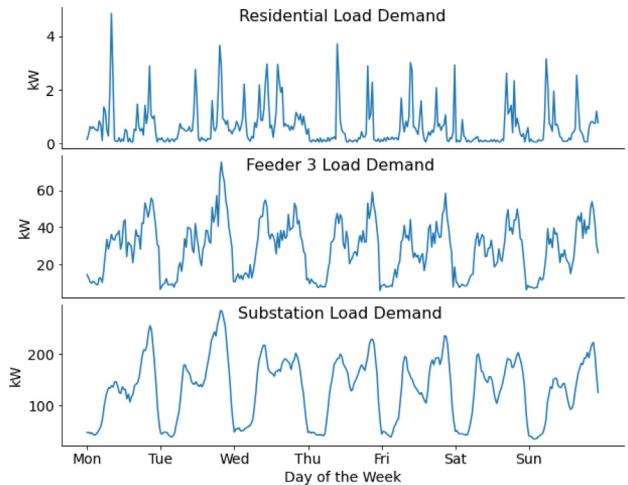


Fig. 11. Example of load demand profile for a week for: residential, feeder, and substation.

Table 13

Number of houses at each feeder.

Feeder number	Numbers of house
Feeder 1	157
Feeder 2	83
Feeder 3	90

load demand is highest for single residential loads, with many sudden peak spikes and irregularities from day to day. On the other hand, load demand is relatively smooth and similar from one day to the next at the highest aggregation level (substation level). Feeders with a relatively high number of customers (such

Table 14

Feeder 3 three phase load forecasting results without FG.

Phase	Model	MAE (kW)	MSE (kW)	RMSE (kW)	MAPE (%)	Training time (s)	Forecasting time (s)
Phase A	ARIMA	3.8337	27.1123	5.2069	0.3742	0.0180	2.7502
	SVM	1.7310	4.4842	2.1176	0.2208	0.0060	0.0180
	CNN	2.054659	7.262564	2.694914	0.231669	0.0160	1.2607
	LSTM	1.5766	4.8692	2.2079	0.2104	0.0265	8.3243
	CNN-LSTM	2.2654	8.3681	2.8928	0.2743	0.0725	30.0154
	Proposed	1.5422	4.8646	2.2057	0.1695	0.0185	2.2394
Phase B	ARIMA	4.5917	35.5890	5.9657	0.3804	0.0150	2.8233
	SVM	1.7695	4.6247	2.1505	0.2346	0.0060	0.0160
	CNN	1.6960	5.5015	2.3455	0.2059	0.0140	1.2379
	LSTM	1.4095	3.7352	1.9327	0.1767	0.0280	8.1895
	CNN-LSTM	1.4807	4.5452	2.1319	0.1824	0.0675	29.1789
	Proposed	1.2817	3.5324	1.8795	0.1567	0.0211	4.8452
Phase C	ARIMA	3.5922	25.3737	5.0372	0.4421	0.0170	1.8645
	SVM	1.8051	4.8119	2.1936	0.2117	0.0070	0.0190
	CNN	2.5627	10.9106	3.3031	0.2620	0.0150	1.9727
	LSTM	1.8965	6.8853	2.6240	0.2908	0.0370	12.2122
	CNN-LSTM	2.1172	8.1158	2.8488	0.2186	0.0770	40.6706
	Proposed	1.6006	5.7488	2.3977	0.1536	0.0206	5.9224

as feeder 3) are much more regular than single residential load demand. Therefore, this research aims to validate the proposed method's ability to be tested on different LV network operation levels, including feeders and substations. For more information about the importance of testing the proposed method at different levels, the author recommends referring to Ref. [Haben et al. \(2023\)](#), a recently published book on this topic.

4.2.1. Feeder forecasting results analysis

In this section, the forecasting results of three phase load demand for the third feeder are chosen to be presented. The presented results have two main cases with and without applying the proposed FG method.

4.2.1.1. Results without FG. [Table 14](#) shows the three-phase load demand forecasting results of the third feeder without applying the proposed FG technique. It provides performance metrics at each phase to compare six models: ARIMA, SVM, LSTM, CNN, CNN-LSTM, and the proposed ECNN-LSTM. It illustrates that ARIMA, SVM, and CNN models have the highest forecasting errors at all phases. The proposed ECNN-LSTM achieved the smallest MAE, MSE, RMSE, and MAPE compared to all other methods at all phases. The ECNN-LSTM proposed method has the best three-phase load forecasting performance and resulted in 32%, 42%, 24%, and 38% improvements in MAE, MSE, RMSE, and MAPE, respectively, at phase-a, over the CNN-LSTM model. Phase-b resulted in 13%, 22%, 12%, and 14% improvements in MAE, MSE, RMSE, and MAPE, respectively, Over the CNN-LSTM model. At phase-c, it resulted in 24%, 29%, 16%, and 30% improvements in MAE, MSE, RMSE, and MAPE, respectively, Over the CNN-LSTM model. Also, the proposed ECNN-LSTM recorded an improvement of 2%, 0.1%, 0.1%, and 19% in MAE, MSE, RMSE, and MAPE, respectively at phase-a, 9%, 5%, 3%, and 11% at phase-b and 16%, 17%, 9% and 47% at phase-c, over the LSTM model.

[Table 14](#) also provides the computational efforts by providing training and forecasting time for all the models. It confirms that both required training and forecasting time is quite small for all models at all phases. Also, it illustrates that the training time for the proposed model was significantly reduced compared to the CNN-LSTM at all phases.

[Fig. 12](#) illustrates the real-time forecasted results of the proposed method in comparison to CNN-LSTM, ARIMA, SVM, CNN, and LSTM models for the three-phase feeder load demand, without the application of the FG technique. It is evident that the deep learning models, including the proposed ECNN-LSTM, CNN-LSTM, CNN, and LSTM, successfully capture the general pattern of the three-phase load profile. However, the ARIMA and SVM

models exhibit the lowest accuracy among all other methods, as confirmed by the figure. While SVM demonstrates slightly better accuracy compared to ARIMA, both models still yield insufficient accuracy for this application. In contrast, the developed ECNN-LSTM model excels in accurately predicting peak and average consumption instances across all phases, surpassing the performance of other models.

4.2.1.2. Results with the FG. [Table 15](#) shows the results of three-phase load demand forecasting at the third feeder by applying the proposed FG technique to all methods. It can be noted that the FG technique has improved the accuracy of all methods compared to the results in [Table 14](#). For instance, the MAE, MSE, RMSE, and MAPE for the ECNN-LSTM method record improvement of 15%, 28%, 13%, and 9% at phase-a, 9%, 17%, 9%, and 0.3% at phase-b, 5%, 8%, 4% and 5% at phase-c, respectively compared to the case without FG technique. On the other hand, the FG application has slightly increased the forecasting errors for both ARIMA and SVM as they are sensitive to the increase in data.

[Table 15](#) also provides the training and forecasting time for all methods. It illustrates that the training method slightly increased because of the application of additional preprocessing, including FG and FS, on all phases compared to the results without FG.

[Fig. 13](#) presents the real-time forecasted results of the proposed method compared to CNN-LSTM, ARIMA, SVM, CNN, and LSTM models for the three-phase feeder load demand, incorporating the FG technique. The accuracy of all deep learning models, including the proposed ECNN-LSTM method, CNN-LSTM, CNN, and LSTM, shows slight improvement compared to the results depicted in [Fig. 12](#). The enhancement is particularly noticeable in CNN-LSTM and CNN models in comparison to LSTM and the proposed ECNN-LSTM as they already exhibit high accuracy even without the FG application.

On the other hand, the application of FG does not provide substantial benefits to the ARIMA and SVM models. In fact, it may slightly reduce their accuracy, confirming their sensitivity to increased data volume, as supported by the results reported in the Tables.

Furthermore, the CNN-LSTM model demonstrates a good improvement in predicting peak values and average consumption across all phases. However, the ECNN-LSTM proposed method outperforms all other models in accurately forecasting peak instances in all phases.

4.2.2. Substation load forecasting analysis

4.2.2.1. Results without the FG. [Table 16](#) shows the results of the three-phase substation load forecasting without applying the

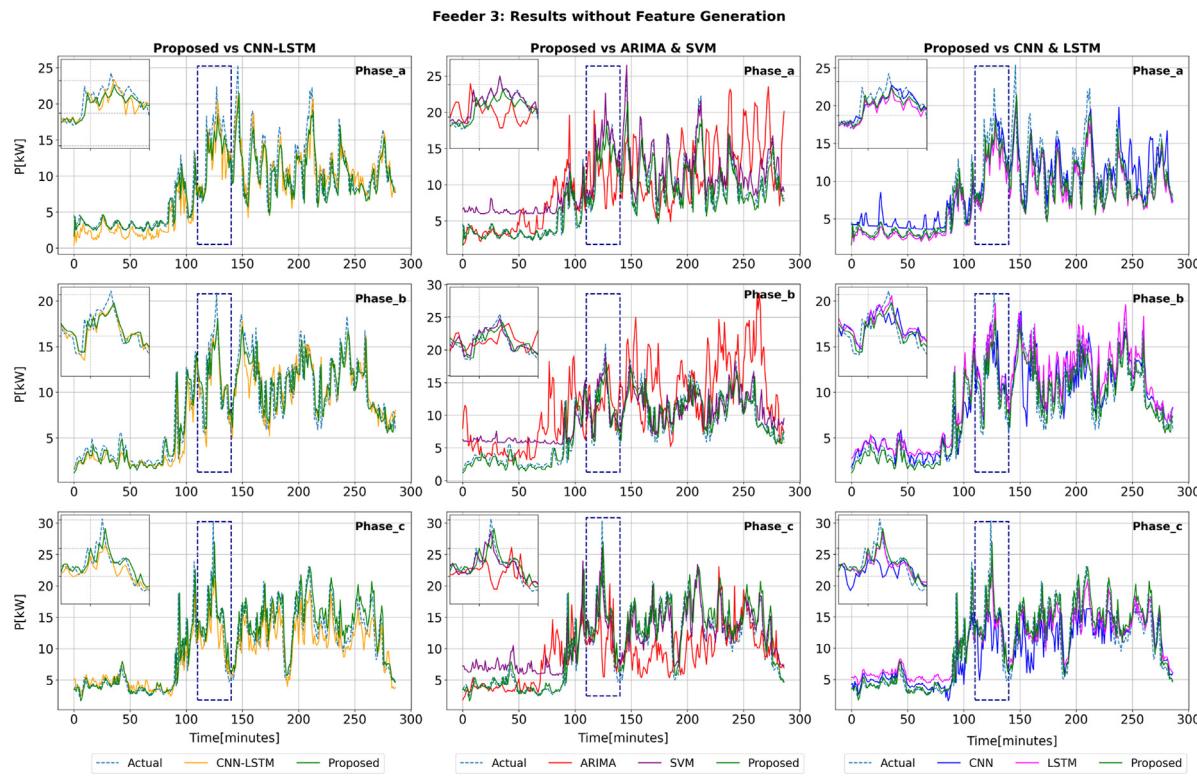


Fig. 12. Real-time forecasting results for the three-phase load demand at Feeder 3 without FG.

Table 15
Three phase load forecasting results with FG at feeder 3.

Phase	Model	MAE (kW)	MSE (kW)	RMSE (kW)	MAPE (%)	Training time (s)	Forecasting time (s)
Phase A	ARIMA	4.0287	28.7473	5.3617	0.3933	0.0175	6.3983
	SVM	2.5084	8.4400	2.9052	0.2767	0.0082	0.0340
	CNN	1.7421	6.0400	2.4576	0.2016	0.0195	2.3401
	LSTM	1.3301	3.8143	1.9530	0.1622	0.0300	13.5886
	CNN-LSTM	1.7668	6.0357	2.4568	0.2005	0.0925	44.6880
	Proposed	1.3094	3.6713	1.9161	0.1540	0.0200	5.9230
Phase B	ARIMA	4.5701	34.8456	5.9030	0.3736	0.0200	6.4249
	SVM	2.0307	5.8372	2.4160	0.2563	0.0100	0.0200
	CNN	1.4872	3.9946	1.9986	0.1765	0.0200	0.7886
	LSTM	1.3326	3.3718	1.8362	0.2361	0.0271	5.8249
	CNN-LSTM	1.3368	3.6057	1.8989	0.1701	0.0700	27.0440
	Proposed	1.1643	2.9211	1.7091	0.1562	0.0230	1.9319
Phase C	ARIMA	3.7959	26.7556	5.1726	0.4823	0.0100	5.5020
	SVM	2.4566	8.9919	2.9987	0.2437	0.0055	0.0151
	CNN	1.8859	7.5719	2.7517	0.1827	0.0183	0.7214
	LSTM	1.5553	5.3169	2.3058	0.1491	0.0200	4.6606
	CNN-LSTM	2.1377	8.2983	2.8807	0.1924	0.0701	21.2633
	Proposed	1.5202	5.2738	2.2965	0.1460	0.0200	1.1368

proposed FG technique. Table 16 illustrates that ARIMA, SVM, and CNN models have the highest forecasting errors compared to other methods at all phases. Table 16 confirms that the proposed ECNN-LSTM has the smallest MAE, MSE, RMSE, and MAPE compared to all other methods at all phases. The ECNN-LSTM proposed method has the best three-phase load forecasting performance and recorded an improvement in MAE, MSE, RMSE, and MAPE, respectively, of 19%, 22%, 12%, and 24% at phase-a, 45%, 66%, 42% and 43.9% at phase-b, 17%, 27%, 15%, and 22%, at phase-c, over the CNN-LSTM model. Also, the proposed ECNN-LSTM recorded an improvement of 23%, 22.9%, 12.2%, and 34% in MAE, MSE, RMSE, and MAPE, respectively, at phase-a, 28%, 31.3%, 17.1% and 36% at Phase b and 7%, 14%, 7% and 17% at phase-c, over the LSTM mode.

Table 16 also illustrates the required computational efforts by providing training and forecasting time for all the models. It shows

that both required training and forecasting time is quite small for all models at all phases, confirming the developed approach's validity. Also, it illustrates that SVM model required the lowest time training and forecasting time, followed by the proposed method. Also, it illustrates that the training time for the proposed model was significantly reduced compared to the CNN-LSTM at all phases.

Fig. 14 shows the real-time forecasted results of the proposed method in comparison to CNN-LSTM, ARIMA, SVM, CNN, and LSTM models for the three-phase substation load demand, without utilizing the FG technique. The findings from this analysis align closely with the observations made in the three-phase forecasting results.

It is evident that all deep learning models, including the proposed hybrid ECNN-LSTM and CNN-LSTM, CNN, and LSTM models, successfully capture the general pattern of the three-phase

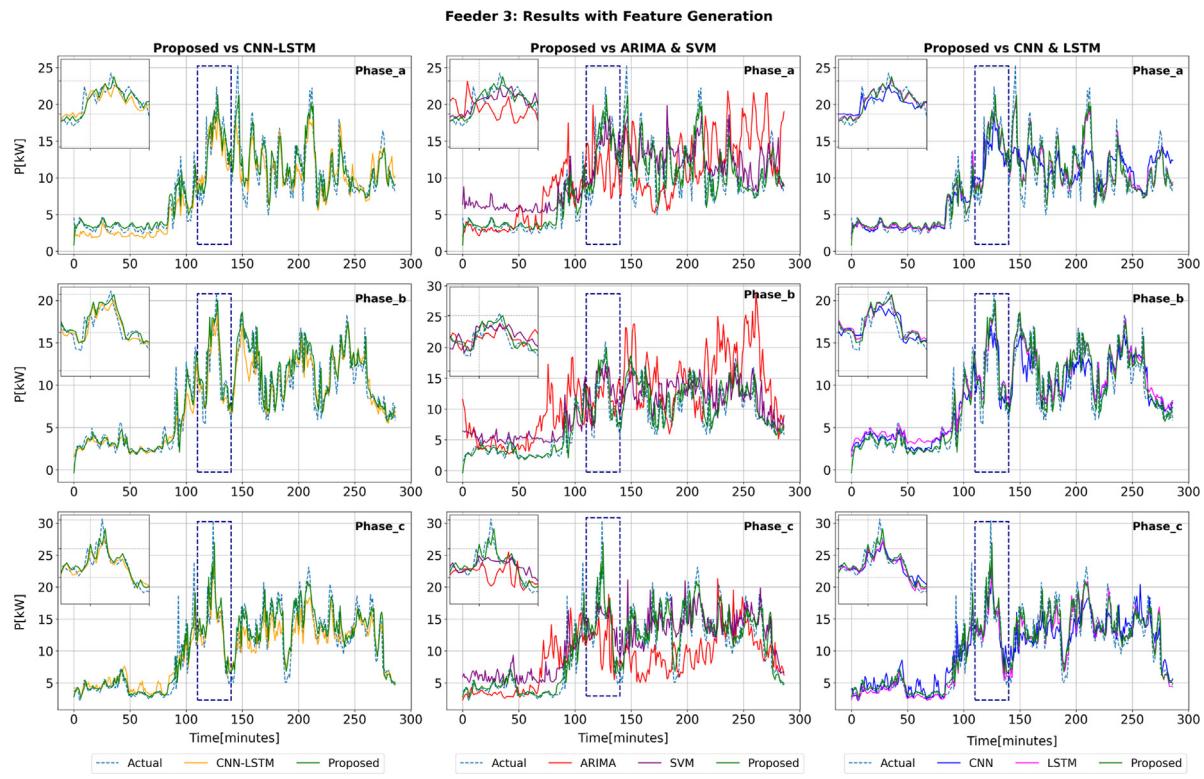


Fig. 13. Real-time forecasting results for the three-phase load demand at Feeder 3 with FG.

Table 16
Three-phase load forecasting results without FG at the substation.

Phase	Model	MAE (kW)	MSE (kW)	RMSE (kW)	MAPE (%)	Training time (s)	Forecasting time (s)
Phase A	ARIMA	14.34466	334.35620	18.28541	0.31656	0.01500	2.42488
	SVM	8.04346	88.39406	9.40181	0.21703	0.00600	0.01500
	CNN	6.7113	69.0956	8.3124	0.1663	0.01400	2.52129
	LSTM	5.8124	49.4920	7.0351	0.1667	0.02600	17.15851
	CNN-LSTM	5.4784	48.7289	6.9806	0.1454	0.07201	60.99028
	Proposed	4.4510	38.1627	6.1776	0.1097	0.01702	4.53232
Phase B	ARIMA	12.3209	250.5888	15.8300	0.2763	0.0100	2.2206
	SVM	7.5633	76.8857	8.7684	0.2199	0.0000	0.0100
	CNN	5.6845	56.9828	7.5487	0.1475	0.0200	0.7462
	LSTM	5.0439	35.7751	5.9812	0.1586	0.0300	4.7456
	CNN-LSTM	6.6134	72.8886	8.5375	0.1797	0.0800	16.9696
	Proposed	3.6561	24.5878	4.9586	0.1009	0.0200	2.0573
Phase C	ARIMA	10.6294	200.4317	14.1574	0.2778	0.0200	2.4303
	SVM	7.9464	81.5670	9.0314	0.2119	0.0000	0.0100
	CNN	6.8596	80.2032	8.9556	0.1708	0.0200	5.3373
	LSTM	4.7988	43.2708	6.5781	0.1362	0.0300	31.8209
	CNN-LSTM	5.3414	51.1011	7.1485	0.1450	0.0900	102.5097
	Proposed	4.4566	37.2161	6.1005	0.1128	0.0300	28.9634

load profile. These results corroborate the findings reported in Table 16, which highlights the superior performance of the proposed method in accurately forecasting peak and average consumption instances across all phases. Furthermore, The LSTM model demonstrates competitive accuracy that is comparable to the proposed method.

In contrast, both SVM and ARIMA exhibit the lowest accuracy among all methods, with significantly higher errors. It is noteworthy, however, that SVM performs relatively better than ARIMA in terms of accuracy, although both models still fall short in comparison to the deep learning approaches.

4.2.2.2. Results with the FG. Table 17 shows the three-phase substation load forecasting results with the application of the proposed FG technique. It can be noted that the FG technique has improved the accuracy of all methods compared to the results

in Table 16. For instance, the MAE, MSE, RMSE, and MAPE for the ECNN-LSTM method record improvement of 14%, 26%, 14%, and 14% at phase-a, 4%, 13%, 7%, and 0.4% at phase-b, 6%, 12%, 6% and 8% at phase-c, respectively compared to the case without FG technique. However, FG application with ARIMA and SVR has slightly increased the furcation errors at all phases.

In addition, Table 17 provides the computational efforts for all methods at all phases. It illustrates that applying the FG method slightly reduced the required training time and training time for the proposed method.

In addition, the results of CNN-LSTM improved significantly as the MAE, MSE, RMSE, and MAPE recorded an improvement of 16%, 9%, 4%, and 22% at phase-a, 38%, 55%, 33%, and 30% at phase-b, 16%, 8%, 4% and 25% at phase-c, respectively compared to the case without FG technique. It can be noticed from Table 17 that CNN-LSTM recorded a greater improvement with the application

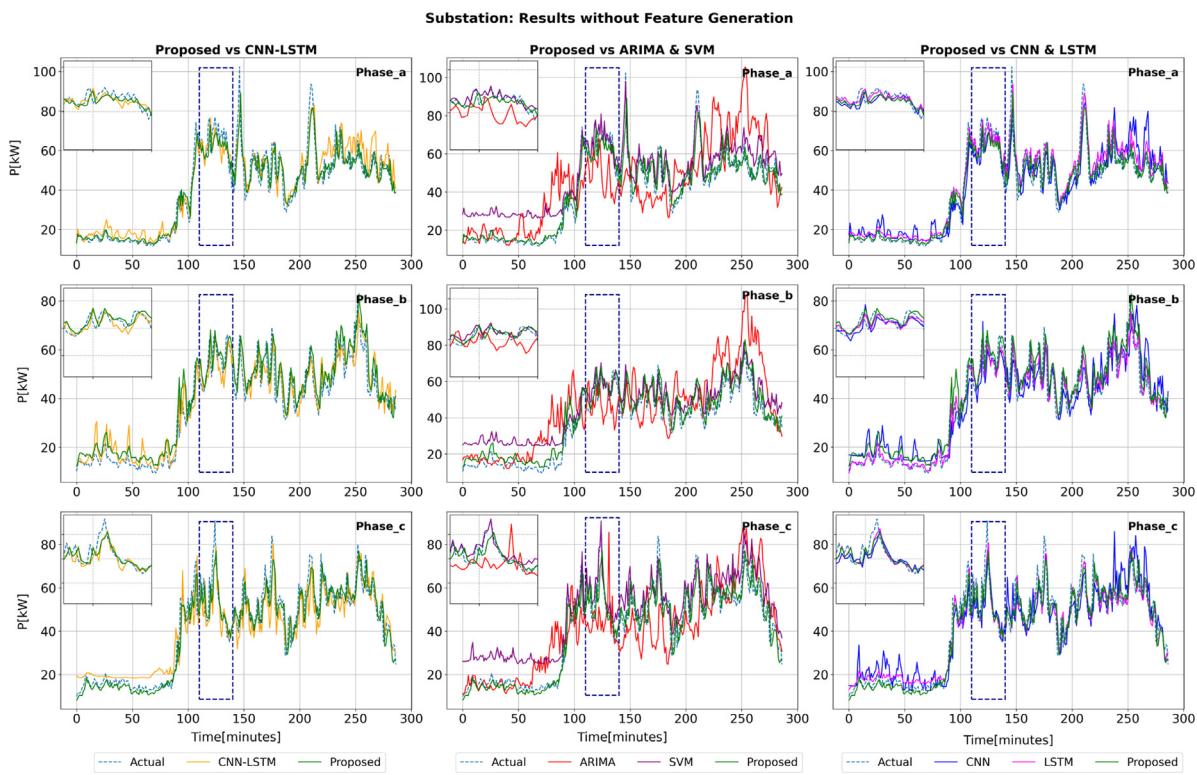


Fig. 14. Real time forecasting results for the three-phase load demand at substation without FG.

Table 17
Three-phase load forecasting results with FG at the substation.

Phase	Model	MAE (kW)	MSE (kW)	RMSE (kW)	MAPE (%)	Training time (s)	Forecasting time (s)
Phase A	ARIMA	14.3978	334.7755	18.2969	0.3180	0.0100	7.9839
	SVM	11.1560	181.0729	13.4563	0.2670	0.0000	0.0200
	CNN	5.3857	53.5131	7.3153	0.1574	0.0100	1.4871
	LSTM	3.8687	29.7188	5.4590	0.0955	0.0200	10.8108
	CNN-LSTM	4.5831	44.5521	6.6747	0.1123	0.0700	44.2071
	Proposed	3.8077	28.3528	5.3178	0.0937	0.0145	2.2087
Phase B	ARIMA	12.0837	241.9886	15.5560	0.2685	0.0200	6.9256
	SVM	9.9538	147.0737	12.1274	0.2610	0.0100	0.0100
	CNN	4.5284	34.6465	5.8861	0.1245	0.0100	1.3983
	LSTM	3.5539	24.3004	4.9295	0.1105	0.0300	8.6012
	CNN-LSTM	4.1150	32.5049	5.7013	0.1252	0.0702	30.5209
	Proposed	3.5278	21.3345	4.6189	0.1005	0.0200	2.1868
Phase C	ARIMA	10.17298	183.285	13.538	0.274343	0.0099	7.611253
	SVM	9.342507	126.516	11.2479	0.238644	0.0100	0.020011
	CNN	4.7682	45.2865	6.7295	0.1122	0.015511	2.200849
	LSTM	4.2019	33.5715	5.8071	0.1135	0.030005	14.02133
	CNN-LSTM	4.5119	40.1265	6.3345	0.1015	0.070024	48.42717
	Proposed	4.1881	32.8340	5.7167	0.1035	0.020002	4.195846

of FG than ECNN-LSTM. The reason is that the CNN-LSTM error rate was much higher than the proposed ECNN-LSTM without the application of FG. Thus, the FG helps the CNN-LSTM to overcome the overfitting that might occur in the first case.

Fig. 15 presents the real-time forecasted results of the proposed ECNN-LSTM model compared to CNN-LSTM, SVM, ARIMA, CNN, and LSTM models for the three-phase substation load demand, with the application of the FG technique. The figure shows a slight improvement in the performance of all deep learning models, including the proposed model, CNN-LSTM, CNN, and LSTM, in forecasting the three-phase load profile with the FG technique.

The CNN-LSTM model exhibits a notable improvement in forecasting the peak value instance and average consumption at all phases compared to the LSTM model and the proposed ECNN-LSTM method. It demonstrates a high level of accuracy even

without the FG technique. However, the SVM and ARIMA models do not show significant improvement with the FG technique, as evidenced by the reported errors in the table and supported by Fig. 15.

4.3. PV generation forecasting analysis

4.3.1. Result without FG

Table 18 presents the results of the PV generation forecasting without applying the proposed FG technique. It compares performance metrics between six models: ARIMA, SVM, LSTM, CNN, CNN-LSTM, and the proposed ECNN-LSTM. It can be noticed that ARIMA, SVM, CNN models recorded the highest errors compared to other models. On the other hand, the proposed ECNN-LSTM achieved the smallest MAE, MSE, RMSE, and MAPE compared to all other methods. The ECNN-LSTM proposed method has the

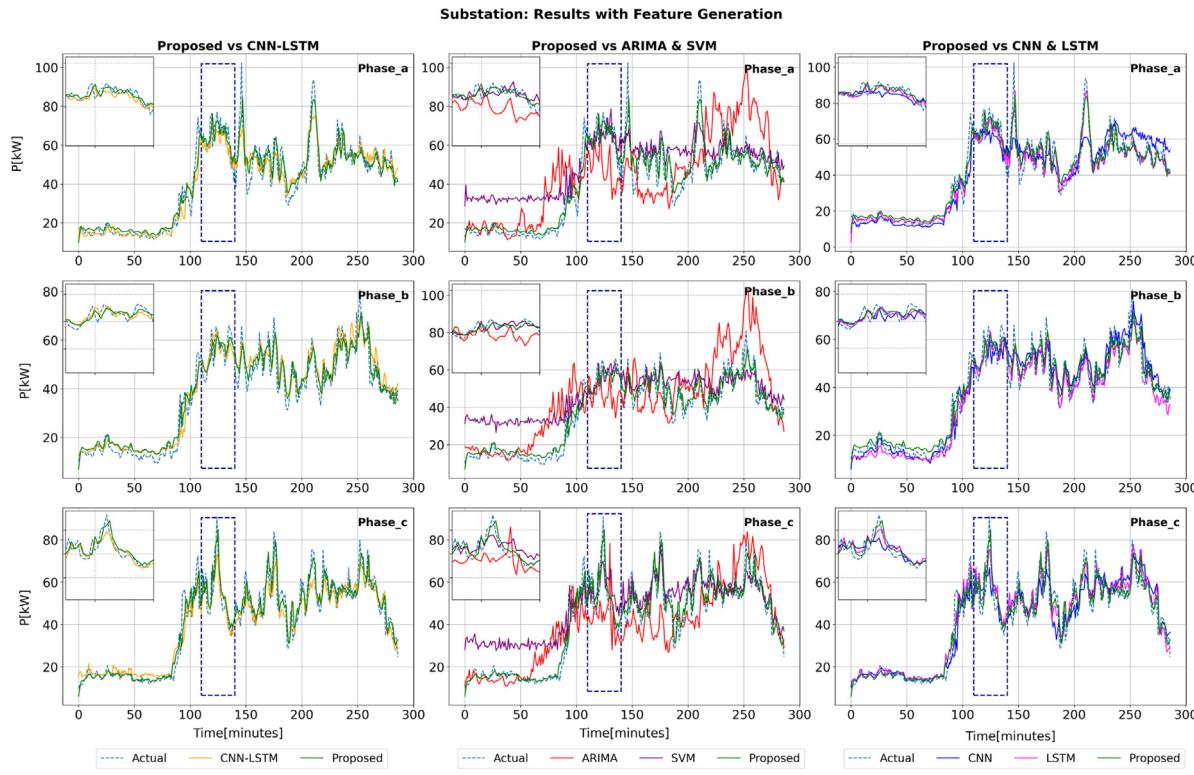


Fig. 15. Real time forecasting results for the three-phase load demand at substation with FG.

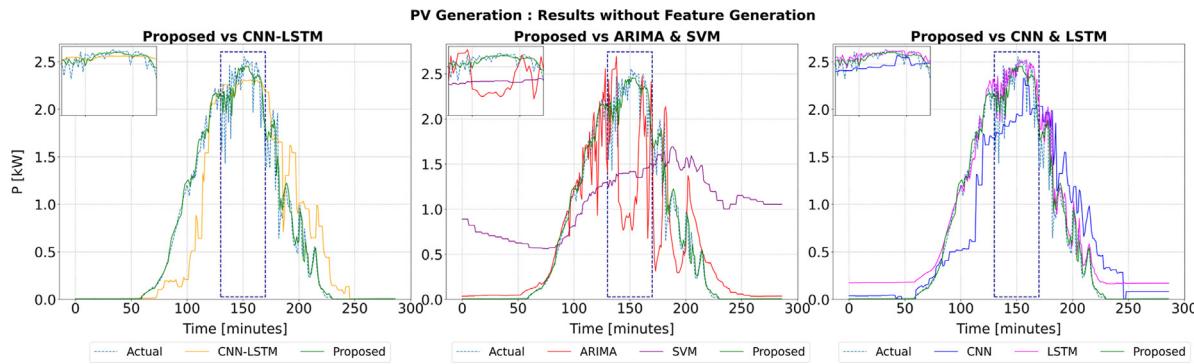


Fig. 16. Real-time forecasting results for the PV generation without FG.

Table 18
PV generation forecasting results without FG.

Method	MAE	MSE	RMSE	MAPE	Prediction time (s)	Training time (s)
ARIMA	0.3856	0.3559	0.5059	0.8422	0.0100	1.5949
SVM	0.7463	0.6567	0.8104	0.7317	0.0100	0.0200
CNN	0.3088	0.1547	0.3933	0.9567	0.0165	1.7576
LSTM	0.0956	0.0289	0.1699	0.6064	0.0200	6.8695
CNN-LSTM	0.2071	0.1047	0.3236	0.5018	0.0910	62.2116
Proposed	0.0924	0.0277	0.1665	0.4960	0.0300	4.5912

best PV generation forecasting performance, resulting in 55%, 74%, 49%, and 1% improvements in MAE, MSE, RMSE, and MAPE, respectively, over the CNN-LSTM model. Also, it recorded 3%, 4%, 2%, and 18% improvements in MAE, MSE, RMSE, and MAPE, respectively, over the LSTM model.

In addition, Table 18 provides the prediction and training time required from all models. It shows that the training time for the

proposed model is significantly lower than that for the CNN-LSTM model.

Fig. 16 presents the real-time forecasted results of the proposed ECNN-LSTM, CNN-LSTM, SVM, ARIMA, CNN, and LSTM models for the PVs generation profile without utilizing the FG technique. The findings presented in Table 18 align with the observations in the figure, emphasizing the superior performance of the proposed method over all other approaches. Notably, the

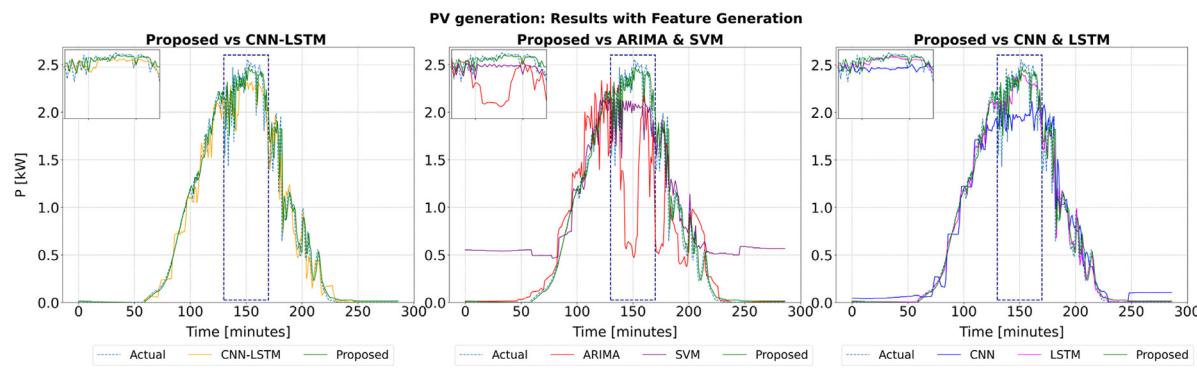


Fig. 17. Real-time forecasting results for the PV generation with FG.

Table 19
PV generation forecasting results with FG.

Method	MAE	MSE	RMSE	MAPE	Prediction time (s)	Training time (s)
ARIMA	0.3523	0.3441	0.5940	0.8122	0.0300	10.0306
SVM	0.4439	0.2410	0.4909	0.6169	0.0100	0.0300
CNN	0.1118	0.0260	0.1613	0.5482	0.0200	6.1732
LSTM	0.1003	0.0306	0.1749	0.4681	0.0334	19.3646
CNN-LSTM	0.1041	0.0246	0.1568	0.4970	0.0800	114.1107
Proposed	0.0825	0.0210	0.1448	0.4549	0.0200	11.1426

SVM method exhibits notably poor forecasting accuracy, further supported by its highest error records among the models. Additionally, the ARIMA model achieves comparatively lower accuracy but outperforms SVM in terms of forecasting.

Furthermore, it is evident from both Table 18 and Fig. 16 that the proposed ECNN-LSTM model demonstrates significantly higher accuracy compared to the CNN-LSTM model. The CNN-LSTM architecture, particularly in Kim and Cho (2019), experiences a low accuracy due to the CNN pooling layer's tendency to discard sensitive information during the learning stage, resulting in less accurate predictions. Moreover, the deeper architecture of the CNN-LSTM model, as opposed to the proposed ECNN-LSTM architecture, introduces excess hidden layers that contribute to overfitting. These findings, supported by Table 18 and Fig. 16, emphasize that adding more hidden layers to the CNN-LSTM architecture increases the number of parameters to train and heightens the risk of overfitting, resulting in less accurate forecasting compared to the ECNN-LSTM model.

4.3.2. Results with FG

Table 19 shows the results of PV generation forecasting by applying the proposed FG technique to all models. It can be noted that the technique has improved the accuracy of all methods compared to the results reported in Table 18. For instance, the MAE, MSE, RMSE, and MAPE for the CNN-LSTM method record improvements of 50%, 77%, 52%, and 1%, respectively, compared to the case without the FG technique. Also, ECNN-LSTM proposed method records an improvement of 11%, 24%, 13%, and 8% in MAE, MSE, RMSE, and MAPE, respectively, over the results without FG. Furthermore, the performance of ARIMA and SVM models is slightly improved with the adoption of FG as the amount of PV generation data is quite small, and it changes within the daytime if compared with load demand data.

It also provides training and prediction time for all models. It illustrates that training and prediction time has increased compared to the results without FG. However, both training and prediction time is quite small, so the impact of time increment is acceptable as applying FG substantially reduces the forecasting errors.

Fig. 17 presents the real-time PV generation forecasted results of the proposed ECNN-LSTM, CNN-LSTM, SVM, ARIMA, CNN, and

LSTM models with the application of the FG technique. A noticeable improvement in accuracy can be observed for all models compared to the results depicted in Fig. 16. Particularly, the CNN-LSTM and CNN models exhibit a significant enhancement in accuracy, indicating that the proposed FG technique effectively addresses the issue of overfitting experienced in the previous case (Fig. 16).

It is worth noting that the ARIMA and SVM models also show a reasonable improvement in performance compared to the results in Fig. 16. This observation suggests that the FG technique helps enhance the forecasting accuracy of ARIMA and SVM models specifically for PV generation. The nature of PV generation data, which is characterized by small-scale variations and a repetitive pattern with zeros during the nighttime, might explain why SVM and ARIMA benefit from the additional features introduced by the FG technique.

5. Conclusion

In conclusion, accurate forecasting in the electric sector is essential due to the growing adoption of RER. This paper proposed a multi-stage deep learning-based framework for short-term forecasting of PV and load. The framework incorporated FG, FS, and OHPT preprocessing techniques to enhance the accuracy of the forecasting methods. An enhanced hybrid CNN-LSTM deep learning model architecture was developed in the final stage of the proposed framework. The framework was evaluated across various DSO scenarios, including multiple single-phase residential loads, three-phase feeders, and secondary substations. The results showed a significant reduction in forecasting errors compared to other leading-edge approaches, highlighting the effectiveness of the proposed framework. For instance, in the case of residential load demand forecasting, the ECNN-LSTM has the best forecasting performance and resulted in 26%, 19%, 10%, and 48% improvements in MAE, MSE, RMSE, and MAPE, respectively, over the CNN-LSTM. This illustrated that the proposed ECNN-LSTM architecture helps reduce forecasting errors by avoiding overfitting, as it has lower hidden layers than the regular CNN-LSTM architecture. Moreover, it showed that applying the developed FG and FS techniques improved the performance of all deep learning

methods. For instance, the application of the FG technique, ECNN-LSTM method records an improvement of 19%, 37%, 21%, and 4% in MAE, MSE, RMSE, and MAPE, respectively, compared to the results without FG in the case of residential load demand. Moreover, applying the FG technique led to a 20% reduction in computational time for both the 100 and 400 households' load forecasting scenarios. Overall, our proposed framework can provide valuable insights for DSO to better manage their local energy systems by accurately predicting short-term load and PV generation.

In addition, deep learning approaches proved to be superior to conventional models, with ARIMA and SVM models showing the highest error rates in all DSO scenarios. However, the application of the FG technique did not significantly improve their performance in residential, three-phase feeders, and substation load forecasting. Conversely, for PV generation forecasting, SVM and ARIMA models exhibited some improvement with the FG technique, taking advantage of the specific characteristics of PV generation data. These results emphasize the effectiveness of deep learning methods in load forecasting, while highlighting the limited applicability of FG for conventional models except in the case of PV generation forecasting.

The proposed framework can be further investigated in future research to explore the potential of incorporating online learning approaches for real-time updates and adaptability to changes in the energy system. Adopting the preprocessing approaches, including FG, FS, and OHPT, is one of the online learning challenges. Nevertheless, a potential solution, such as re-learning the model parodically, which requires a powerful computing effort, can be investigated in further research. Additionally, the framework's applicability can be explored in larger-scale power systems with diverse renewable energy sources, and the model's performance can be evaluated for longer-term forecasting horizons. Furthermore, the impact of incorporating other external factors, such as the adoption of electric vehicles and demand response programs, on forecasting accuracy can also be investigated.

CRediT authorship contribution statement

Mohammad Ahmad A. Al-Ja'afreh: Conceptualization, Methodology, Software, Visualization, Data curation, Validation, Writing – review & editing. **Geev Mokryani:** Conceptualization, Validation, Writing – review & editing. **Bilal Amjad:** Conceptualization, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Geev Mokryani reports financial support was provided by UK Research and Innovation.

Data availability

Datasets related to this article can be found at <https://data.mendeley.com/datasets/zympd537wv/1>, an open-source online data repository hosted at Mendeley Data (Al-Ja'afreh, Mokryani, Amjad, 2023).

Acknowledgment

This work was supported in-part by Innovate UK GCRF Energy Catalyst Pi-CREST project under Grant number 41358, and in-part by Mut'ah University, Jordia.

References

- Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M., 2019. Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2623–2631.
- Al-Musayl, M.S., Deo, R.C., Adamowski, J.F., Li, Y., 2018. Short-term electricity demand forecasting with MARS, SVR and ARIMA models using aggregated demand data in queensland, Australia. *Adv. Eng. Inform.* 35 (2017), 1–16. <http://dx.doi.org/10.1016/j.aei.2017.11.002>.
- Alqatawneh, I., Deng, R., Rabeyee, K., Chao, Z., Gu, F., Ball, A.D., 2021. A developed convolutional neural network architecture for condition monitoring. In: 2021 26th International Conference on Automation and Computing. ICAC, pp. 1–6.
- Alqatawneh, I., Rabeyee, K., Zhang, C., Feng, G., Gu, F., Ball, A.D., 2020. A modified activation function for deep convolutional neural network and its application to condition monitoring. In: International Conference on Maintenance Engineering, pp. 895–909.
- Aouad, M., Hajj, H., Shaban, K., Jabr, R.A., El-Hajj, W., 2022. A CNN-sequence-to-sequence network with attention for residential short-term load forecasting. *Electr. Power Syst. Res.* 211 (June), 108152. <http://dx.doi.org/10.1016/j.epsr.2022.108152>.
- Aprililia, H., Yang, H.-T., Huang, C.-M., 2020. Short-Term Photovoltaic Power Forecasting Using a Convolutional Neural Network-Salp Swarm Algorithm, doi:<http://dx.doi.org/10.3390/en13081879>.
- Awad, M., Khanna, R., 2015. Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers. Springer nature.
- Bennett, C.J., Stewart, R.A., Lu, J.W., 2014. Forecasting low voltage distribution network demand profiles using a pattern recognition based expert system. *Energy* 67, 200–212.
- Bera, S., Shrivastava, V.K., 2020. Effect of pooling strategy on convolutional neural network for classification of hyperspectral remote sensing images. *IET Image Process.* 14 (3), 480–486. <http://dx.doi.org/10.1049/iet-ipr.2019.0561>.
- Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B., 2011. Algorithms for hyper-parameter optimization. *Adv. Neural Inf. Process. Syst.* 24.
- Bergstra, J., Yamins, D., Cox, D.D., 2013. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In: Proceedings of the 12th Python in Science Conference, Vol. 13, p. 20.
- Boyd, G., 2017. SPEN - DSO vision. CIRED - Open Access Proc. J. 2017 (1), 2007–2010. <http://dx.doi.org/10.1049/oap-cired.2017.1044>.
- Budin, L., Duilo, I., Delimar, M., 2022. Day-ahead multiple households load forecasting using deep learning and unsupervised clustering. In: 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology. MIPRO, pp. 30–35. <http://dx.doi.org/10.23919/MIPRO55190.2022.9803401>.
- Dai, Y., Wang, Y., Leng, M., Yang, X., Zhou, Q., 2022. LOWESS smoothing and random forest based GRU model: A short-term photovoltaic power generation forecasting method. *Energy* 256, 124661. <http://dx.doi.org/10.1016/J.ENERGY.2022.124661>.
- du Plessis, A.A., Strauss, J.M., Rix, A.J., 2021. Short-term solar power forecasting: Investigating the ability of deep learning models to capture low-level utility-scale photovoltaic system behaviour. *Appl. Energy* 285, 116395. <http://dx.doi.org/10.1016/J.APENERGY.2020.116395>.
- Elahe, M.F., Jin, M., Zeng, P., 2022. Knowledge-based systematic feature extraction for identifying households with plug-in electric vehicles. *IEEE Trans. Smart Grid* 13 (3), 2259–2268. <http://dx.doi.org/10.1109/TSG.2022.3146556>.
- Fekri, M.N., Patel, H., Grolinger, K., Sharma, V., 2021. Deep learning for load forecasting with smart meter data: Online adaptive recurrent neural network. *Appl. Energy* 282 (PA), 116177. <http://dx.doi.org/10.1016/j.apenergy.2020.116177>.
- Goude, Y., Nedellec, R., Kong, N., 2013. Local short and middle term electricity load forecasting with semi-parametric additive models. *IEEE Trans. Smart Grid* 5 (1), 440–446.
- Haben, S., Arora, S., Giassimidis, G., Voss, M., Vukadinović Greetham, D., 2021. Review of low voltage load forecasting: Methods, applications, and recommendations. *Appl. Energy* 304 (October), 117798. <http://dx.doi.org/10.1016/j.apenergy.2021.117798>.
- Haben, S., Voss, M., Holderbaum, W., 2023. Core Concepts and Methods in Load Forecasting: With Applications in Distribution Networks. Springer Nature.
- Han, Y., Tang, B., Deng, L., 2019. An enhanced convolutional neural network with enlarged receptive fields for fault diagnosis of planetary gearboxes. *Comput. Ind.* 107, 50–58.
- He, K., Sun, J., 2015. Convolutional neural networks at constrained time cost. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5353–5360.
- Huang, C.-J., Kuo, P.-H., 2019. Multiple-input deep convolutional neural network model for short-term photovoltaic power forecasting. *IEEE Access* 7, 74822–74834. <http://dx.doi.org/10.1109/ACCESS.2019.2921238>.
- Jahangir, H., Tayarani, H., Gougheri, S.S., Golkar, M.A., Ahmadian, A., Elkamel, A., 2020. Deep learning-based forecasting approach in smart grids with micro-clustering and bidirectional LSTM network. *IEEE Trans. Ind. Electron.* 68 (9), 8298–8309.

- Kabilan, R., et al., 2021. Short-term power prediction of building integrated photovoltaic (BIPV) system based on machine learning algorithms. *Int. J. Photoenergy* 2021, 5582418. <http://dx.doi.org/10.1155/2021/5582418>.
- Kim, T.-Y., Cho, S.-B., 2018. Web traffic anomaly detection using C-LSTM neural networks. *Expert Syst. Appl.* 106, 66–76.
- Kim, T.-Y., Cho, S.-B., 2019. Predicting residential energy consumption using CNN-LSTM neural networks. *Energy* 182, 72–81.
- Kim, Y., Seo, K., Harrington, R.J., Lee, Y., Kim, H., Kim, S., 2020. High accuracy modeling for solar pv power generation using noble bd-lstm-based neural networks with ema. *Appl. Sci.* 10 (20), 1–16. <http://dx.doi.org/10.3390/app10207339>.
- Korkmaz, D., Acikgoz, H., Yildiz, C., 2021. A novel short-term photovoltaic power forecasting approach based on deep convolutional neural network. <http://dx.doi.org/10.1080/15435075.2021.1875474>.
- Laouafi, A., Mordjaoui, M., Haddad, S., Boukelia, T.E., Ganouche, A., 2017. Online electricity demand forecasting based on an effective forecast combination methodology. *Electr. Power Syst. Res.* 148, 35–47.
- Li, C., Tao, Y., Ao, W., Yang, S., Bai, Y., 2018. Improving forecasting accuracy of daily enterprise electricity consumption using a random forest based on ensemble empirical mode decomposition. *Energy* 165, 1220–1227.
- Li, P., Zhou, K., Lu, X., Yang, S., 2020. A hybrid deep learning model for short-term PV power forecasting. *Appl. Energy* 259, 114216. <http://dx.doi.org/10.1016/j.apenergy.2019.114216>.
- Ma, Y., Lv, Q., Zhang, R., Zhang, Y., Zhu, H., Yin, W., 2021. Short-term photovoltaic power forecasting method based on irradiance correction and error forecasting. *Energy Rep.* 7, 5495–5509. <http://dx.doi.org/10.1016/j.egyr.2021.08.167>.
- McKenna, E., Thomsonand, M., Barton, J., 2015. CREST demand model. <http://dx.doi.org/10.17028/rd.lboro.2001129.v8>.
- Mishra, M., Byomakesha Dash, P., Nayak, J., Naik, B., Kumar Swain, S., 2020. Deep learning and wavelet transform integrated approach for short-term solar PV power prediction. *Measurement* 166, 108250. <http://dx.doi.org/10.1016/j.measurement.2020.108250>.
- Motepet, S., Hasan, A.N., Stopporth, R., 2019. Improving load forecasting process for a power distribution network using hybrid AI and deep learning algorithms. *IEEE Access* 7, 82584–82598.
- Nijhuis, M., Gibescu, M., Cobben, J.F.G., 2015. Assessment of the impacts of the renewable energy and ICT driven energy transition on distribution networks. *Renew. Sustain. Energy Rev.* 52, 1003–1014. <http://dx.doi.org/10.1016/j.rser.2015.07.124>.
- Nirthika, R., Manivannan, S., Ramanan, A., Wang, R., 2022. Pooling in convolutional neural networks for medical image analysis: a survey and an empirical study. *Neural Comput. Appl.* 34 (7), 5321–5347. <http://dx.doi.org/10.1007/s00521-022-06953-8>.
- Parvez, I., Sarwat, A., Debnath, A., Olowu, T., Dastgir, M.G., Riggs, H., 2020. Multi-layer perceptron based photovoltaic forecasting for rooftop PV applications in smart grid. In: 2020 SoutheastCon. pp. 1–6. <http://dx.doi.org/10.1109/SoutheastCon44009.2020.9249681>.
- Paszke, A., et al., 2019. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* 32.
2014. Photovoltaic (PV) solar panel energy generation data.
- Razavi, S.E., Arefi, A., Ledwich, G., Nourbakhsh, G., Smith, D.B., Minakshi, M., 2020. From load to net energy forecasting: Short-term residential forecasting for the blend of load and PV behind the meter. *IEEE Access* 8, 224343–224353. <http://dx.doi.org/10.1109/ACCESS.2020.3044307>.
- Richardson, I., Thomson, M., Infield, D., Clifford, C., 2010. Domestic electricity use: A high-resolution energy demand model. *Energy Build.* 42 (10), 1878–1887. <http://dx.doi.org/10.1016/j.enbuild.2010.05.023>.
- Sajjad, M., et al., 2020. A novel CNN-gru-based hybrid approach for short-term residential load forecasting. *IEEE Access* 8, 143759–143768.
- Shi, H., Xu, M., Li, R., 2017. Deep learning for household load forecasting—A novel pooling deep RNN. *IEEE Trans. Smart Grid* 9 (5), 5271–5280.
- Sorour, A., Fazeli, M., Monfared, M., Fahmy, A.A., Searle, J.R., Lewis, R.P., 2021. Forecast-based energy management for domestic PV-battery systems: a UK case study. *IEEE Access* 9, 58953–58965.
- Su, H., Liu, F., Xie, Y., Xing, F., Meyyappan, S., Yang, L., 2015. Region segmentation in histopathological breast cancer images using deep convolutional neural network. In: 2015 IEEE 12th International Symposium on Biomedical Imaging. ISBI, pp. 55–58.
- Takeda, H., Tamura, Y., Sato, S., 2016. Using the ensemble Kalman filter for electricity load forecasting and analysis. *Energy* 104, 184–198. <http://dx.doi.org/10.1016/j.energy.2016.03.070>.
- Tan, M., Hu, C., Chen, J., Wang, L., Li, Z., 2022. Multi-node load forecasting based on multi-task learning with modal feature extraction. *Eng. Appl. Artif. Intell.* 112 (2021), 104856. <http://dx.doi.org/10.1016/j.engappai.2022.104856>.
- Trivedi, R., Patra, S., Khadem, S., 2022. A data-driven short-term PV generation and load forecasting approach for microgrid applications. *IEEE J. Emerg. Sel. Top. Ind. Electron.*
- UK Power Networks, 2015. UK power networks, validation of photovoltaic (PV) connection assessment tool: Closedown report, no. March.
- Valgaev, O., Kupzog, F., Schmeck, H., 2017. Building power demand forecasting using K-nearest neighbours model – practical application in smart city demo aspenn project. *CIRED - Open Access Proc. J.* 2017 (1), 1601–1604. <http://dx.doi.org/10.1049/oap-cired.2017.0419>.
- VanDeventer, W., et al., 2019. Short-term PV power forecasting using hybrid GASVM technique. *Renew. Energy* 140, 367–379. <http://dx.doi.org/10.1016/j.renene.2019.02.087>.
- Wang, Q., Ji, S., Hu, M., Li, W., Liu, F., Zhu, L., 2018. Term photovoltaic power generation combination forecasting method based on similar day and cross entropy theory. <http://dx.doi.org/10.1155/2018/6973297>.
- Wang, J., Zhu, W., Zhang, W., Sun, D., 2009. A trend fixed on firstly and seasonal adjustment model combined with the ε -SVR for short-term forecasting of electricity demand. *Energy Policy* 37 (11), 4901–4909. <http://dx.doi.org/10.1016/j.enpol.2009.06.046>.
- Yang, A., Li, W., Yang, X., 2019. Short-term electricity load forecasting based on feature selection and least squares support vector machines. *Knowl.-Based Syst.* 163, 159–173.
- Yin, M., Li, K., Yu, J., 2022. A data-driven approach for microgrid distributed generation planning under uncertainties. *Appl. Energy* 309, 118429.
- Zang, H., et al., 2018. Hybrid method for short-term photovoltaic power forecasting based on deep convolutional neural network. *IET Gener. Transm. Distrib.* 12 (20), 4557–4567. <http://dx.doi.org/10.1049/iet-gtd.2018.5847>.
- Zeiler, M.D., Fergus, R., 2014. Visualizing and understanding convolutional networks. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September (2014) 6–12, Proceedings, Part I 13. pp. 818–833.
- Zhang, W., Peng, G., Li, C., Chen, Y., Zhang, Z., 2017. A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. *Sensors* 17 (2), 425.
- Zhang, J., Wei, Y.-M., Li, D., Tan, Z., Zhou, J., 2018. Short term electricity load forecasting using a hybrid model. *Energy* 158, 774–781. <http://dx.doi.org/10.1016/j.energy.2018.06.012>.