Integrated Multi-Head Self-Attention Transformer model for electricity demand prediction incorporating local climate variables

Sujan Ghimire a, Thong Nguyen-Huy b, Mohanad S. AL-Musaylh c, Ravinesh C. Deo a, David Casillas-Pérez d, e, Sancho Salcedo-Sanz a, e

a School of Mathematics, Physics and Computing, University of Southern Queensland, Springfield, QLD, 4300, Australia
b Centre for Applied Climate Sciences, University of Southern Queensland, Toowoomba, 4350, QLD, Australia
c Department of Information Technologies, Management Technical College, Southern Technical University, Basrah 61001, Iraq
d Department of Signal Processing and Communications, Universidad Rey Juan Carlos, Fuenlabrada, 28942, Madrid, Spain
e Department of Signal Processing and Communications, Universidad de Alcalá, Alcalá de Henares, 28805, Madrid, Spain

HIGHLIGHTS

• Electricity forecast model considers lagged power demand and local climatic variable.
• Multi-Head Self Attention Transformer-based artificial intelligence model is proposed.
• The model generates electricity prediction intervals at 95% confidence.
• The model supersedes the performance of state-of-the-art deep learning models.
• Forecast intervals with probabilistic prediction provide insights for market analysis.

Article Info

Dataset link: https://www.energex.com.au
MSC:
0000
1111

Keywords:
Electricity demand forecasting
Sustainable energy
Artificial Intelligence
Deep Learning
Transformer Networks
Kernel Density Estimation

ABSTRACT

This paper develops a trustworthy deep learning model that considers electricity demand (G) and local climate conditions. The model utilizes Multi-Head Self-Attention Transformer (TNET) to capture critical information from G, to attain reliable predictions with local climate (rainfall, radiation, humidity, evaporation, and maximum and minimum temperatures) data from Energex substations in Queensland, Australia. The TNET model is then evaluated with deep learning models (Long-Short Term Memory LSTM, Bidirectional LSTM BiLSTM, Gated Recurrent Unit GRU, Convolutional Neural Networks CNN, and Deep Neural Network DNN) based on robust model assessment metrics. The Kernel Density Estimation method is used to generate the prediction interval (PI) of electricity demand forecasts and derive probability metrics and results to show the developed TNET model is accurate for all the substations. The study concludes that the proposed TNET model is a reliable electricity demand predictive tool that has high accuracy and low predictive errors and could be employed as a stratagem by demand modelers and energy policy-makers who wish to incorporate climatic factors into electricity demand patterns and develop national energy market insights and analysis systems.
1. Introduction

Clean and affordable energy remains a critical component of Sustainable Development Goal 7 of the United Nations Development Program (UNDP 2018–2021). The UNDP’s goal is to promote renewable and efficient energy technologies that are inclusive and responsive to population needs in urban and rural areas, for women, men, households and businesses. These technologies, that include the development of models for future prediction of the energy demand, can assist in strategic energy planning decisions to ensure sustainable and equitable supply of energy to support livelihoods and develop regional areas, inaccessibility to electricity supply, reduce poverty, improve health and well-being, and create resilient societies. Additionally, accurate energy supply–demand models can also help understand the role of various causal factors such as climate or weather variables, to help manage conventional energy generations, re-balance the additional supply of renewable energies, such as solar and wind power to reduce greenhouse gas emissions, lessen the burden of air pollution and further contribute towards sustainable development [1].

A major challenging issue in the energy industry is providing accurate predictions of future electricity demand \( G \) data to make informed decisions for smooth operation of the electricity power supply and demand system. Several factors can affect the \( G \) data, including the population growth, economic development, industrialisation, and climate variables, that vary for local regions, country to country. Changing atmospheric variables such as temperature, rainfall, vapour pressure, humidity, and evaporation, for example, can affect the amount of electricity needed in a given sub-station zone. Australia suffers from extreme weather episodes as a result of climate change, such as droughts, local high temperatures, floods, and bush-fires [2]. Consequently, the Australian electricity demand is more likely to be impacted by climate change over long-term or weather patterns over short term periods since the electricity demand data are associated with cooler and hotter days, respectively \([3,4]\). This problem, as well as the non-linearity and the non-stationarity issues associated with climatic variables \([5,6]\) affecting the \( G \) data makes it necessary to have a robust estimation model of electricity demand considering local weather or climate data.

Deep learning (DL) models have been extensively developed in areas of classification \([7]\), clustering \([8]\), image analysis \([9]\), and regression problems. Numerous research studies have demonstrated DL algorithms’ superiority over other methods. First, these approaches do not require instructions or manual feature production. Instead, they instruct the user on how to interpret the data \([10,11]\). With large datasets, DL methods provide new strategies to identify the best input variables and hidden patterns throughout the development stage \([4,12,13]\). In contrast to machine learning (ML) methods, DL has been identified as a faster method to optimise thousands of iterations and hyperparameters in significantly short time \([14,15]\). In addition, DL has been used for a variety of real life prediction applications, many of them related to renewable and sustainable energy, such as wind speed forecasting \([16,17]\) or solar radiation prediction \([18,19]\), among others.

A few of the most widely used DL architectures include Convolutional Neural Networks (CNN), Long-Short Term Memory (LSTM) Networks (DNN), Deep Neural Networks (DNN), Recurrent Neural Networks (RNN), Radial Basis Function Networks (RBFN), Gated Recurrent Units (GRU), Generative Adversarial Networks (GAN) and Multilayer Perceptrons (MLP) \([20]\). In 2017, the work of \([21]\) developed a powerful model called the Transformer Network, which has been used in image generation, text generation, and audio summarisation since then. As mentioned in \([22]\), the use of the transformer models has increased with different application problems, for example, speech processing, computer vision and natural language processing due to its transform ability that has been used to pre-trained data. Further, a modified transformer model known as a deep hybrid single-Head Self-Attention Transformer (TNET), has proven to be a modern high-quality model given that: (i) it is computationally efficient, (ii) more parallelizable \([21]\) and (iii) has a wide usage in time-series prediction \([23–29]\). Within the TNET model, the mechanism of self-attention can be used to extract information that are contained in a time series dataset. To pursue this challenge, the encoder–decoder architecture aims to convert the input variable-length sequence into a transitional fixed-length sequence, which is then decoded to generate the output fixed-length sequence. The key benefit of any TNET model is its “recall” and global feature mapping as main requirements to build a DL model \([30]\). In the study of \([31]\), the authors conducted experiments on real time-series data (local climatological dataset, Geo-Magnetic field dataset and Qinghuayuan Tunnel shield dataset) using the transformer model to demonstrate that this method is able to make efficient predictions. In \([32]\), a transformer model was proposed for network traffic prediction where the result demonstrated that the proposed model can capture long time-series features and can parallelise the output results. A transformer model was also used to predict wind speed in five different US locations to determine its superiority over standalone deep learning models such as LSTM, GRU, RNN, and Backpropagation Neural Network (BPNN) \([28]\). In the study of \([33]\), a transformer model was proposed for maximum tracking of power in photovoltaic cells. The results of this study indicated that a model learns the temporal patterns in the time-series data quickly when it is equipped with transformer modules with an attention mechanism. Using a testing dataset gathered over 200 consecutive hours, the transformer model achieved a prediction with a Mean Average Percentage Error of 0.47% on non-zero operational voltage points, resulting in an average power efficiency of 99.54% and a peak power efficiency of 99.98%. In the study of \([34]\), a transformer model forecasted solar PV power generation one-hour-ahead to outperform the compared models such as an ANN, LSTM, and GRU methods.

Recently, in the study of \([35]\), an experiment was performed on six datasets using energy, traffic, economics, weather and disease variables while using the Frequency Enhanced Decomposed Transformer model. According to the results, compared to state-of-the-art approaches, the proposed transformer model was able to minimise the prediction error by 14.8% for multivariate time series and 22.6% for univariate time series. Finally, in the study of \([36]\), a new hybrid renewable energy forecasting system combining traditional linear and state-of-the-art ML models such as Seasonal Autoregressive Integrated Moving Average (SARIMA) and Transformer Neural Network (TNN) was proposed using exogenous data and validated on the five datasets with wind speed and solar energy. In comparison to hybrid models such as SARIMA+RNN, SARIMA+LSTM, SARIMA+GRU and standalone models such as SARIMA, RNN, LSTM and GRU, the Transformer consistently produced the best results. One major challenge of the transformer model, however, is to capture the temporal patterns at different time scales, optimise self-attention mechanisms, and handle static covariates within the predictor-target dataset, in order to produce accurate predictions of electricity demand. The study presents a deep hybrid Multi-Head Self Attention Transformer model (TNET), which is an enhanced framework of Transformer-based time-series predictions. The multi-head attention, instead of single-head, employed in the proposed model enables one to map both longer- and shorter-term sequence dependencies differently \([37–44]\), and therefore, is the primary contribution of this research work.

This paper presents the TNET model that is trained, validated and tested as an objective model to improve power demand point predictions based on electricity demand and local climate variables from Energex, Australia and Scientific Information for Land Owners (SILO) repository, respectively. In order to demonstrate the significant capability of the TNET model, the objective model was compared with five different deep learning (LSTM, BILSTM, CNN, GRU, DNN) models. Based on how predictions are expressed, \( G \) predictions can be classified as deterministic or probabilistic. With \( G \) being highly uncertain and volatile, probabilistic prediction provides decision-makers with a practical and useful reference by providing a prediction interval based
on various levels of confidence. Therefore, in this study firstly the point prediction (deterministic prediction) was estimated using the TNET and the comparative models and then the non-parametric Kernel Density Estimation (NKDE) based on Gaussian Distribution was employed to fit the point Prediction Error (PE) to generate the Prediction Interval (PI) (Probabilistic prediction) at a level of 95% confidence. The paper is divided into five sections: an introduction, a literature review, a work contribution, a theoretical framework for deep learning algorithms and theories, including TNET, LSTM, BLSTM, GPR, CNN, and DNN, a predictive model development for the study six models, a discussion of the results of the models, and a conclusion explaining the benefits and findings of the research.

2. Theoretical framework of deep learning models

Deep Learning (DL) approaches are particularly powerful as can extract attributes from data without preprocessing. Traditional methods of Machine Learning (ML) involve tedious tasks such as preprocessing, feature extraction, and segmentation, resulting in inferior efficiency and accuracy. Therefore, this study compares the proposed hybrid DL model with standalone models. We now discuss the DL-based objective model, known as the Multi-Head Self-Attention Transformer (TNET) and the benchmark models: Long-Short Term Memory Networks (LSTM), Deep Neural Networks (DNN), Gated Recurrent Units (GRUs), Convolutional Neural Networks (CNNs) and Bidirectional LSTMs (BLSTM).

2.1. Deep Neural Network

DNNs are ML techniques that are derived from the structures and functions of the human brain [45,46]. A DNN can perform both unsupervised and supervised operations including feature extraction and classification, and transformation by observing, analysing, learning, and making judgements for complicated issues [47]. The key difference between DNN and ANN is the size and complexity, which are two distinguishing characteristics between the two neural network approaches. An ANN typically contains three layers of input, hidden, and output. Each layer’s nodes or neurons are linked to those in neighbouring layers, with each link having a weight. For each node, input variables are multiplied by their weights and summed together [48]. In this process, the sum of products is transformed by activation functions, such as sigmoid or hyperbolic tangents. The ANN model can be mathematically formulated as:

\[ Y = F \left( \sum_{j=1}^{m} W_{ij} \cdot F \left( \sum_{k=1}^{n} W_{kj} X_k + B_j \right) + B_i \right) \tag{1} \]

where: \( W_{ij} \) and \( W_{kj} \) denote the weightings between input and hidden and between hidden and output layers, respectively; \( X_i \) is the input variable; \( m \) and \( n \) correspond to the number of neurons in each hidden and input layer; \( B_j \) and \( B_i \) are respective to the bias values of the neurons in the hidden and output layers and \( F \) signifies the activation function and \( Y \) refers the output.

The optimal weightings are determined through a process known as the neural network training. In contrast with ANNs, DNN is trained by iteratively changing the weights in the network to minimise the difference between observed and predicted values. More specifically, for feature extraction and transformation, DNN models employ several non-linear hidden layers, each of which may have a different functional capability (Fig. 1(a)).

2.2. Convolutional Neural Network

In 1990, LeChun et al. proposed the Convolutional Neural Network (CNN), which is a multilayer perceptron [49]. The CNN analyzes input data primarily using local connections and parameter sharing that reduces the likelihood of overfitting [50]. Additionally, CNNs provide many advantages over typical neural networks, including high input distortion tolerance, parallelism, self-learning, and rapid training speeds. A number of fields have used CNN in the past, including computer vision and object tracking, fault detection and identification, medical image classification, activity recognition, and writer identification. As a result, it has been useful in the modelling of time-series with prescribed observations (e.g. electricity demand, solar radiation, wind speed) [51]. For this type of datasets, CNN models learns from a series of previous patterns to predict the next value in the sequence [52]. An input layer of a CNN is followed by a convolutional layer, a pooling layer, and a fully linked classification layer (see Fig. 1(b)). This layer’s units receive input from the preceding layer. Convolution between each filter and input produces an input map. The method involves moving one function and summing dot products. The input-kernel convolution is defined by Eqs. (2) and (3):

\[ f_i (n) = x(n) k(1) + x(n−1) k(2) + \cdots + x(0) k(n) \tag{2} \]

\[ f_i (n) = \sum_{j=n}^{m} x(j + 1) k(m − j + 1) \tag{3} \]

where \( x \) and \( k \) signify the input vector and kernel filter with the lengths of \( n \) and \( m \), respectively.

The CNN architecture includes a pooling layer for each convolutional operation (See Fig. 1(b)) where the signal is down-sampled using the pooling layers. Typically, the maximum pooling strategy and the average pooling strategy are used. In this procedure, the input signal is separated into multiple pooling segments where the average value computed from each segment is:

\[ F (n) = \frac{1}{n} \sum_{i=1}^{n} f_i \tag{4} \]

In this study, a 1-D filter is applied in order to convolution the time series prediction using the 1-D CNN algorithm. The 1-D CNN model is based on the traditional 2-D CNN, and therefore both algorithms operate similarly. The feature signal input of layer \( l \) during forwards propagation is the convoluted output from the previous feature signal of layer \( (l−1) \) and the individual filter kernel. Given each \( i \) comprises an \( m' \) feature signal, this step can be mathematically expressed as in Eq. (5):

\[ y_{k, l}^{(i)} = b_{i,k}^{(l)} + \sum_{j=1}^{m'} \text{conv1D} \left( w_{i,j}^{(k,l)}, x_{l−1}^{(i)} \right) \tag{5} \]

where \( y_{k, l}^{(i)} \) denotes the \( k \)th feature signal input, \( b_{i,k}^{(l)} \) represents the bias of the \( k \)th signal, \( w_{i,j}^{(k,l)} \) signifies the kernel weight from \( l \)th feature signal at layer \( (l−1) \) to \( k \)th signal on the layer \( l \), and \( x_{l−1}^{(i)} \) means the \( i \)th feature signal output on layer \( (l−1) \). The \( k \)th signal output is acquired by implementing the activation function to Eq. (5).

A collection of reduced features is acquired after the convolution and pooling processes. This data is then fed into a Multi-layer Perceptron (MLP), which is fully connected, and an activation function, which carries out the regression process for prediction task [18,53].

2.3. Long-Short Term Memory Network

By using a purpose-built LSTM memory cell, the Long-Short Term Memory Network (LSTM) is an improved form of the classic Recurrent Neural Network (RNN). Compared to typical RNNs, the LSTM design provides the user with an adequate algorithm for addressing the vanishing gradient problem associated with RNN [19]. LSTM uses the current input and the prior recurrent neurone states to maintain the current recurrent neurone states even though the recurrent units are replaced by multiple memory cells, making it possible to represent long-term dependencies in sequential datasets [54,55]. LSTM cells incorporate four gating units, which include input, output, forgetting gates, and self-recurrent memory units, in order to manage the interactions of
tanh (and sigmoid activation function expressed as below:

\[ y_t = \sigma(W_x h_{t-1} + b_x, y_t), \]

while \( \tanh(h_t) \) and \( \sigma(h_t) \) correspond to the weights of the forward and backward direction respectively. Given LSTM with the input and targeted output of as \( x = (x_1, x_2, \ldots, x_T) \) and \( y = (y_1, y_2, \ldots, y_T) \), respectively, the targeted output \( y_t \) calculated at time \( t \) can be obtained through Eqs. (6) to (12) [19,57].

\[ f_t = \sigma(W_f h_{t-1} + b_f) \]  
\[ i_t = \sigma(W_i h_{t-1} + b_i) \]  
\[ c_t = \tanh(W_c h_{t-1} + b_c) \]  
\[ o_t = \sigma(W_o h_{t-1} + b_o) \]  
\[ h_t = o_t \tanh(c_t) \]  
\[ y_t = W_y h_t + b_y \]  

where \( f_t, i_t, c_t, \) and \( o_t \) are forget, input, current cell, and output gate with their respective weight matrices \( W_f, W_i, W_c, W_o \) and bias vectors \( b_f, h_i, h_c, b_o \) and \( b_y \). Both \( h_t \) and \( h_{t-1} \) denote the current and previous hidden state output, respectively. The term \( x_t \) refers to the input vector. The symbol \( * \) means the element-wise multiplication while \( \tanh(\cdot) \) and \( \sigma(\cdot) \) represent the corresponding hyperbolic tangent and sigmoid activation function expressed as below:

\[ \sigma(x) = \frac{1}{1 + e^{-x}} \]  
\[ \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \]

2.4. Bi-directional LSTM

As LSTMs evaluate input vectors only in one direction, significant feature information can be dropped during training, which prevents the sequence information from being fully evaluated [58,59]. In order to address this issue, the BILSTM model incorporates a bidirectional structure to collect both forward and backward data representations (Fig. 2). Both forward and backward propagation directions are covered by the BILSTM [60]. As a result of the forward and backward LSTM layers, one can determine the past and future of an input sequence. The BILSTM produces a final output vector \( y_i \) by:

\[ y_i = \sigma(W_{fb} h_i + W_{bh} h_{i-1} + b_i) \]

In the forward direction, the information in \( h_i \) from past time-series values is stored in internal state while in the backward direction, information is held in \( h_{i-1} \) from future sequence values. The \( W_{fb} \) and \( W_{bh} \) correspond to the weights of the forward and backward direction from the internal unit to the output while \( b_i \) and \( b_f \) signifies the bias vector and activation function (set to sigmoid or linear functions) of the output layer, respectively.

2.5. Gated Recurrent Unit

It is well known that the clockwork RNN, Depth-Gated LSTM, and Gated Recurrent Unit (GRU) are some variants of LSTM that are created by merging and modifying gating techniques. The GRU in particular, is a popular LSTM variants that has been extensively used in the literature [18]. As a result, this study used the GRU as one of its benchmark models. In GRU, forget and input gates are combined into a single update gate (\( r_{ij} \)), as well as cell state and hidden state. In contrast to LSTM, GRU has two gates called update and reset gates [61]. As a result of the update gate (\( r_{ij} \)), information is transferred from earlier time steps to subsequent time steps, while the reset gate (\( r_{ij} \)) determines what information is erased from earlier time steps.

If we denote the input and the targeted output of GRU as \( x = (x_1, x_2, \ldots, x_T) \) and \( y = (y_1, y_2, \ldots, y_T) \), respectively, the targeted output \( y_t \) at time \( t \) can be computed using Eqs. (16) to (20) [62].

\[ z_t = \sigma(W_z [h_{t-1}, x_t] + b_z) \]  
\[ h_t = o_t \tanh(c_t) \]  
\[ y_t = W_y h_t + b_y \]  
\[ y_t = \sigma(W_y h_t + b_y) \]
the past regardless of the length of the series. This capability allows different weights. The transformer, unlike RNN, can reach any point in parameter that is commonly assumed to have 6 layers [21]. Decoding is with a fixed length. In each encoder stack, the encoder number is a free the encoder converts the input sequence into an intermediate sequence supported by encoders and decoders. To produce the output sequence, can transform input sequences into variable-length output sequences, summarisation, and image generation. The implementation of attention 2.6. Multi-Head Self-Attention Transformer Model

The transformer architecture model is not only popular among machine translation researchers, but also for audio, music, text summarisation, and image generation. The implementation of attention and self-attention mechanisms eliminates the need for recurrent neural networks [69]. As with sequence-to-sequence structures, transformers can transform input sequences into variable-length output sequences, supported by encoders and decoders. To produce the output sequence, the encoder converts the input sequence into an intermediate sequence with a fixed length. In each encoder stack, the encoder number is a free parameter that is commonly assumed to have 6 layers [21]. Decoding is carried out using a stack of decoders with the same layer numbers as the encoder (Fig. 3). As each encoding layer uses different settings, it uses different weights. The transformer, unlike RNN, can reach any point in the past regardless of the length of the series. This capability allows the transformer to identify long-term dependencies and eliminate the vanishing gradient issue associated with RNNs. Further, unlike RNN, the transformer runs in parallel at high speed [64]. In a transformer model, the attention mechanism allows the neural network to focus on inputs that are most relevant to the current output [65]. Self-attention involves comparing a single sequence with itself in order to enhance representations [66]. As well as increasing the interpretability of input and output sequences, the self-attention mechanism can be used to select informative features. Self-attention records time-series sequential information regardless of length, unlike basic attention, which learns representations without understanding the time-series [67]. The correlation of the input data itself can be captured by mapping one length-varied sequence of inputs \{x_1, x_2, ..., x_T\} to another equal length sequence \{z_1, z_2, ..., z_T\} to acquire the global correlation between the input and output. The Eq. (21) is used to describe the attention mechanism and Eq. (22) to achieve the attention weight scores from the input data using a scaled Dot Product.

\[
C(Q, K, V) = \text{softmax}(f(Q, K)) \cdot V
\]

\[
f_{\text{SDV}}(Q, K) = \frac{(Q \cdot K^T)}{\sqrt{d_k}}
\]

where \(Q, K,\) and \(V\) are the respective query, key, and value matrices, the three components (i.e., \(Q, K,\) and \(V\)) are analogous to systems that retrieve information in which the matching key and its value are found via a query, and \(d_k\) represents the dimension of \(V\) employed to scale the weight to suppress the dot product values from being too large.

The transformer model includes single-head attention (SHA) and multi-head attention (MHSAA) [68]. SHA helps model time series variations by determining attention weights (Fig. 3(b)). In MHSAA models, the attention module repeats its computations in parallel. In the MHSAA module, the individual attention outputs are concatenated and linearly transformed (Fig. 3(c)). MHSAA also maps long-term and short-term sequence dependencies differently and is defined as follows:

\[
\text{MHSAA}(Q, K, V) = [H_1 \ldots H_{mH}] W_H
\]

\[
H_h = \text{Attention}(Q W_Q^{(h)}, K W_K^{(h)}, V W_V^{(h)})
\]

where \(W_Q^{(h)}, W_K^{(h)},\) and \(W_V^{(h)}\) are head-specific weights corresponding to keys, queries, and values and \(W_H^{(h)}\) is a linear combination of outputs linked all heads \(H_h\).

2.7. Uncertainty prediction

Using point prediction, only one predicted value is provided for one target value, without any indication of the likelihood of the prediction being accurate. However, Prediction Intervals (PIs) provide both a range of targets that can be covered and a measure of their accuracy, also known as coverage probability. This study expresses uncertainty as two quantiles (i.e., PI) from the underlying distribution of Prediction Errors (PE). Based on a number of earlier studies, NKDE is reported to be more reliable and sensitive than Quantile Regression Method [69]. Therefore, NKDE was used to quantify the PIs at a 95% confidence level. To fit a probability density function (PDF) universally from \(K\) and bandwidth \((b)\), the NKDE relies on the kernel function \((K)\) and bandwidth \((h)\). For a group of \(G\) Prediction Error \((PE)\) data \(e = \{e_1, e_2, ..., e_n\}\), \(n\) is the number of \(PE\) samples. Based on the NKDE principle, we estimate the PDF of the daily \(G\) prediction error in Eq. (25).

\[
f(e, h) = \frac{1}{nh} \sum_{i=1}^{n} K \left( \frac{e - e_i}{h} \right)
\]

where \(f(e, h)\) is the PDF of \(K\) at \(PE\) sample of \(e\) in which \(K\) signifies a symmetric unimodal PDF with 0 at the centre and the bandwidth \(h\) specifies the interval division for the error data distribution while \(e_i\)
3. Predictive model development

In this section, we discuss the electricity demand data and other climate input parameters used for modelling, followed by a procedure for preprocessing the data. Next, we define the data pre-processing stages, Deep Learning models, and interval prediction procedure. Python 3.8 [71] was used to implement data pre-processing, predictive modelling, and assessment on a Windows 10 platform with Intel® core™ TМ and i9 Generation 10 with 3.8 GHz central processing unit and 32 GB of memory. Furthermore, the model uses the MATLAB (MATrix LABoratory) program R2020b for statistical analysis and displays. The Python language provides public libraries based on DL capabilities whereas Tensor Flow [72], Keras [73], and Scikit-learn [74] are examples of essential and unique libraries. We present a three-phase framework including data pre-processing, a Multi-Head Self-Attention Transformer (TNET) model, and interval prediction. In order to improve prediction outcomes, input data is pre-processed and normalised. As part of the second phase, a TNET model and other DL-based benchmark models are developed to estimate daily electricity demand. Using Nonparametric Kernel Density Estimation (NKDE), prediction intervals are calculated from point prediction residual error.

3.1. Data collection

In this study, 30-minute interval electricity demand (G) data from Energex((https://www.energex.com.au)) for four substations (Coopers Plains, Browns Plains, Beenleigh and Bethania; Fig. 4) in South-east Queensland, Australia, over July 2011–June 2021, are used. Extracted electricity demand data from Energex had a total of 280 561 records at 30-min intervals, which were down sampled to 3640 records for daily electricity demand prediction. This was accomplished using Eq. (31), where \( H_n \) is a function that takes a collection of electricity demand data as input and down samples it to a specified time, and \( n \) denotes the down sampling rate (i.e., for daily transformation, \( n = 48 \)). The descriptive statistics are shown in Table 1.

\[
H_n = \sum_{j=n}^{j=n+1} H_{ji}
\]  

(31)

Fig. 5(a) shows the distribution of \( G \) by day of the week for all four substations. As can be seen, Coopers Plains, Browns Plains, and Beenleigh have much larger variations of \( G \) distributions during the week than on weekends. Bethania, on the other hand, has a similar distribution of \( G \) throughout the week. Similarly, a box plot depicted in Fig. 5(b) presents the distribution of all \( G \) for month. As we can see from the seasonal behaviour, the variances of the \( G \) distributions in December and February (the summer season) are significantly larger than those in other months.
In addition, a variety of predictor attributes were collected according to local climatic data, including rainfall, radiation, humidity, evaporation, and maximum and minimum temperatures. Data were taken from the Scientific Information for Land Owners (SILO) patched point data repository for the Bureau of Meteorology weather station [75]. These data are found on Long Paddock website: https://www.longpaddock.qld.gov.au/silo/ and are now detailed in Table 2.

![Map of the Electricity sub-stations showing the locations.](image)

**Table 1**

<table>
<thead>
<tr>
<th>Statistical parameters</th>
<th>Coopers plains</th>
<th>Browns plains</th>
<th>Beenleigh</th>
<th>Bethania</th>
</tr>
</thead>
<tbody>
<tr>
<td>Median (MW)</td>
<td>458.38</td>
<td>990.37</td>
<td>326.82</td>
<td>314.31</td>
</tr>
<tr>
<td>Mean (MW)</td>
<td>410.10</td>
<td>1003.68</td>
<td>334.03</td>
<td>324.68</td>
</tr>
<tr>
<td>Standard deviation (MW)</td>
<td>129.07</td>
<td>135.61</td>
<td>106.21</td>
<td>52.62</td>
</tr>
<tr>
<td>Variance</td>
<td>16658.51</td>
<td>18390.23</td>
<td>11281.36</td>
<td>2769.20</td>
</tr>
<tr>
<td>Maximum (MW)</td>
<td>693.70</td>
<td>1640.36</td>
<td>610.99</td>
<td>599.02</td>
</tr>
<tr>
<td>Minimum (MW)</td>
<td>111.83</td>
<td>145.32</td>
<td>0.00</td>
<td>44.97</td>
</tr>
<tr>
<td>Range</td>
<td>693.70</td>
<td>1640.36</td>
<td>610.99</td>
<td>599.02</td>
</tr>
<tr>
<td>Interquartile range</td>
<td>213.53</td>
<td>160.26</td>
<td>158.11</td>
<td>60.23</td>
</tr>
<tr>
<td>Skewness</td>
<td>-0.90</td>
<td>0.63</td>
<td>-0.41</td>
<td>1.08</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>4.87</td>
<td>4.52</td>
<td>2.69</td>
<td>5.31</td>
</tr>
</tbody>
</table>

**Table 2**

<table>
<thead>
<tr>
<th>Predictor variables from SILO</th>
<th>Acronym</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum temperature (°C)</td>
<td>Tmax</td>
</tr>
<tr>
<td>Minimum temperature (°C)</td>
<td>Tmin</td>
</tr>
<tr>
<td>Vapour pressure (hPa)</td>
<td>VP</td>
</tr>
<tr>
<td>Vapour pressure deficit (hPa)</td>
<td>VPD</td>
</tr>
<tr>
<td>Evaporation - synthetic estimate (mm)</td>
<td>Esyn</td>
</tr>
<tr>
<td>Solar radiation - total incoming</td>
<td>GSR</td>
</tr>
<tr>
<td>Relative humidity at the time of maximum temperature (%)</td>
<td>Rhmax</td>
</tr>
<tr>
<td>Relative humidity at the time of minimum temperature (%)</td>
<td>Rhmin</td>
</tr>
<tr>
<td>Evapotranspiration - Morton’s areal actual evapotranspiration (mm)</td>
<td>Evap</td>
</tr>
<tr>
<td>Mean sea level pressure (hPa)</td>
<td>MSLP</td>
</tr>
</tbody>
</table>

**3.2. Data pre-processing**

Following the acquisition of $G$ and local climate data for each sub-station operated by Energex, Australia, the first steps towards deploying the predictive models for point and P1 generation are data pre-processing and partitioning strategies. In the pre-processing steps, the $G$ data in single dimension is converted to multi-dimension using the delay embedding method to generate appropriate inputs for predictive model training. In this method, higher-dimensional data can be compressed within a single-dimensional chaotic data [76]. In this case, the higher dimensional $G$ time-series data can be acquired as follows:

$$G_p(t) = [G(t), G(t − r), ..., G(t − (d − 1)r)]$$

where $d$ and $r$ represent the respective embedding dimension and time delay and the time delay ($r$) is determined based on the Mutual Information Test (MIF) [77] where optimum $r$ is the first minimum of the MIF.

Further, the false nearest model [78] was employed to get the optimum $d$. Fig. 6(a) shows the time delay($r$) of $G$ time series for four sub-stations. For Coopers Plains sub-stations, $r$ decreases quasi-exponentially until 2 days later and then reaches a minimum at 6 days after that. Thus, we chose the latter as the time delay for the Coopers Plains. Similarly for the Browns Plains,Beenleigh and Bethania the first minimum occur at 6,5 and 11, respectively. Based on the MIF criterion in Fig. 6(a), the most optimal inputs for $G$ prediction at four sub-stations can be mathematically represented by Eqs. (33) to (36):  

Coopers Plains: $G_t = f(G_{t−1}, G_{t−2}, G_{t−3}, G_{t−4}, G_{t−5}, G_{t−6})$  

(Brown Plains: $G_t = f(G_{t−1}, G_{t−2}, G_{t−3}, G_{t−4}, G_{t−5})$  

(Bethania: $G_t = f(G_{t−1}, G_{t−2}, G_{t−3}, G_{t−4}, G_{t−5})$  

In addition, local climatic data from SILO and the historical lagged variable of $G$ are normalised using min–max normalisations to reduce
calculation burdens in later model runs. Mathematically, the range of normalised data is $[0,1]$ expressed by Eq. (37).

$$x^* = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}$$  

(37)

where, $x^*$ is the normalised value of $x$, $x_{\text{max}}$ and $x_{\text{min}}$ are maximum and minimum value of $x$ respectively.

The prediction result is inverse normalised to ensure that the predicted value of the network output is the same as the true value, and the inverse normalisation function is as follows.

$$\hat{y}_i = y_i \cdot (x_{\text{max}} - x_{\text{min}}) + x_{\text{min}}$$  

(38)

where $y_i$ is the model output value and $\hat{y}_i$ is the predicted value.

After normalisation of data the lagged data of $G$ and climate variables are merged to get the input and target matrix for the predictive models (e.g. Eqs. (39) and (40) for Coopers Plains and Browns Plains).

$$\text{Input} = \left\{ G_{t-1}, G_{t-2}, G_{t-3}, G_{t-4}, G_{t-5}, T_{\text{max}}, T_{\text{min}}, V_p, V_{PD}, E_{\text{syn}}, GSR_t, R_h_{\text{max}}, R_h_{\text{min}}, E_{\text{|n|}}, MSLP_t \right\}$$  

(39)

$$T\text{arget} = \left\{ G_t \right\}$$  

(40)

where $G_t$ is the current electricity demand, $G_{t-1}$, $G_{t-2}$, $G_{t-3}$, $G_{t-4}$ are lagged values of electricity demand, $T_{\text{max}}$, $T_{\text{min}}$, $V_p$, $V_{PD}$, $E_{\text{syn}}$, $GSR_t$, $R_h_{\text{max}}$, $R_h_{\text{min}}$, $E_{\text{|n|}}$, $MSLP_t$, are the current values of the Maximum Temperature, Minimum Temperature, Vapour pressure, Vapour Pressure Deficit, Solar Radiation, Relative Humidity at Maximum Temperature, Relative Humidity at Maximum Temperature, Morton’s Areal Actual Evapotranspiration and Mean Sea level Pressure at current time ($t$) respectively.

Finally, pre-processed data is divided into training, validation, and testing sets. For training, data from 01/07/2011 through 30/06/2021 is used, while for testing, data from 01/07/2021 through 30/06/2022 is used. The training data is used to build the models, and the validation data is used to select the system parameters that perform best on the training data. Our model construction and parameter selection do not include testing data. Moreover, we used 0.2 validation splits, which means 20% of training data are used for validation. For daily $G$ prediction, 2804 data points are used, and for validation and testing, 471 and 365, respectively. For training, validation, and testing, the input matrix for Coopers Plains and Browns Plains sub-stations is $2801 \times 16$, $471 \times 16$ and $365 \times 16$ (i.e. 2801 rows of 16 features, 6 lagged $G$, and 10 climate variables from SILO). Similarly for Beenleigh $2801 \times 15$, $471 \times 15$ and $365 \times 15$ are used for training, validation and testing respectively whereas for the Bethania sub-station, $2801 \times 21$, $471 \times 21$ and $365 \times 21$ for training, validation and testing respectively.

In addition, this study has evaluated feature importance using Categorical Boosting Regression, primarily to investigate the relationship between inputs (lagged $G$ and local climate variables) and the target ($G$). The important value is represented by the vertical axis on the left,
S. Ghimire et al.

Fig. 5b. Monthly distribution of $G$ in the four substations week (a) Coppers Plains, (b) Browns Plains, (c) Beenleigh, and (d) Bethania.

while the accumulation value of feature importance is represented by the vertical axis on the right. As shown in Fig. 6(b), the first lag of $G$ (i.e. $G(t - 1)$) has an important role in $G$ prediction, contributing 35%, 32%, 39% and 40% to respective substations Coopers Plains, Browns Plains, Beenleigh, and Bethania, respectively. The second lag of $G$ (i.e. $G(t - 2)$) is the second most important predictor for Coopers Plains and Beenleigh sub-stations, whereas $T_{max}$ is the second most important predictor for Browns Plains and Bethania sub-stations.

3.3. Multi-Head Self-Attention Transformer Model Framework

To demonstrate the capability of the objective model for daily electricity demand $G$ prediction, the Multi-Head Self-Attention Transformer Model (TNET) has been developed. The proposed TNET model captures the temporal relationships among input sequences by embedding time vectors $[79]$. It is, however, impossible for the transformer model to arrange each input sequence temporally without time embeddings. Thus, this study adopts Time2Vec($t_2v$) $[80]$ to create a set of time embedded vectors. In fact, the $t_2v$ is an orthogonal yet complementary technique that employs learnable weights to realise the vector representation of time, allowing it to be readily integrated with the transformer model whereas the $t_2v$ should be time rescaling invariant consisting of both periodic and non-periodic patterns.

For a time series dataset denoted as $T$, the vector representation is as follows:

$$t_2v(T)[i] = \begin{cases} \omega_i T + \varphi, & i = 0 \\ F(\omega_i T + \varphi), & 1 \leq i \leq k \end{cases}$$ (41)

where the non-periodic feature is described by a linear function with slope $\omega$ and intercept $\varphi$.

When the periodic wave lies within $[1,k]$ condition, a periodic feature involves linear function wrapped within a function $F$. In addition, $\sin$ function is chosen for a periodic feature as it produces the best performance accuracy $[80]$. The computation of the input sequence pre-processing using the $t_2v$ embedding matrix is shown in Algorithm 1 whereby periodic and non-periodic vectors are computed and integrated into a time-dependent sequence of the proposed TNET model inputs.

Algorithm 1 Time2Vec Embedding Algorithm

Input: $x$ : Input Matrix, $weight\_linear, bias\_linear$:Weights and Bias Matrices for Non-periodic, $weight\_periodic, bias\_periodic$. Weights and bias Matrices for periodic

Output: output: Output vector

1: Function $Time2Vec(x,weights\_linear,bias\_linear,weights\_periodic, bias\_periodic)$
2: $time\_linear \leftarrow weight\_linear \times x + bias\_linear$
3: $time\_periodic \leftarrow \sin(multiply(x,weights\_periodic) + bias\_periodic)$
4: $output \leftarrow concat([time\_linear, time\_periodic])$
5: end

The single head self-attention (SA) is acquired based on the Algorithm 2. To create the query, key, and value matrices, inputs are passed into three dense layers. The outputs are fed into Eq. (22) to derive the attention weights. As shown in Algorithm 3, the input sequence is
processed through every \( n \) SA modules, and the outputs are integrated together, which is used as the inputs for a linear transformation through the dense layer of the MHSA module before being tuned to minimise transformer encoder loss.

In this study, there are three MHSA, each with 12 heads. More insight into \( G \) time-series variations is provided by increasing the number of heads. Long historical data dependencies can be also efficiently modelled using three such multi-head modules. Fig. 7 illustrates the transformer encoder and decoder.

In brief, the encoder unit contains the multi-head self-attention (MHSA) model accompanied with dropout, layer normalisation and conv1D (1D-CNN). Rectified Linear Unit (ReLU) are used as activation function for the CNN layer. The encoder’s input is the input sequence vectorised by the \( t_2 v \). The encoder’s fixed length output is then sent to the decoder that includes global average pooling, dropout, and dense output. Eventually, the output layer consists of only one node corresponding to the length of output window and is activated by utilising a linear function, which was required for regression issues to generate a numerical prediction value rather than a probability.

This study further utilise the Hyperopt library 0.1.2 [81], which is a Python package to optimise hyperparameters (i.e., filters, hidden units in Dense layer and Batch size) of the proposed TNET model to deduce the optimal hyperparameter for best performance. The ideal hyperparameters are marked at the positions where the Root Mean Square Metrics shows the best performance in validation data. Table 3 shows the considered search space for each of these hyperparameters. Hyperopt runs through all of the potential combinations of these values to find the best one.

<table>
<thead>
<tr>
<th>Algorithm 2 Single-Head Attention Mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input: ( \text{inputs} ) : Input Matrix, query, key, value : Query, Key and Value Dense Layers, ( d, k ) : dimensions of key</td>
</tr>
<tr>
<td>Output: ( \text{attn}_{\text{out}} ) : Output Attention</td>
</tr>
<tr>
<td>1: Function Single-Head-Attention(( \text{inputs}, \text{query}, \text{key}, \text{value}, d, k ))</td>
</tr>
<tr>
<td>2: ( \text{query} \leftarrow \text{query}(\text{inputs}) )</td>
</tr>
<tr>
<td>3: ( \text{key} \leftarrow \text{key}(\text{inputs}) )</td>
</tr>
<tr>
<td>4: ( \text{attn}_{\text{weights}} \leftarrow \text{matmul}(\text{q, k, transpose_b = True}) )</td>
</tr>
<tr>
<td>5: ( \text{attn}<em>{\text{weights}} \leftarrow \text{map}\text{fn}(\lambda x: x/\sqrt{d_k}, \text{attn}</em>{\text{weights}}) )</td>
</tr>
<tr>
<td>6: ( \text{attn}<em>{\text{weights}} \leftarrow \text{softmax}(\text{attn}</em>{\text{weights}}) )</td>
</tr>
<tr>
<td>7: ( v \leftarrow \text{value}(\text{inputs}) )</td>
</tr>
<tr>
<td>8: ( \text{attn}<em>{\text{out}} \leftarrow \text{matmul}(\text{attn}</em>{\text{weights}}, v) )</td>
</tr>
<tr>
<td>9: end</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm 3 Multi-Head Attention Mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input: ( \text{input} : ) Input Matrix, ( n_{\text{heads}} ) : Number of Single-Head Attention modules, ( \text{attn}_{\text{heads}} ) : Single head Attention modules, ( \text{linear} ) : Dense Layer</td>
</tr>
<tr>
<td>Output: ( \text{multi}_{\text{linear}} ) : Output Attention of Multiple Single Head Attention Modules</td>
</tr>
<tr>
<td>1: Function Multi-Head-Attention(( \text{inputs}, n_{\text{heads}}, \text{attn}_{\text{heads}}, \text{linear} ))</td>
</tr>
<tr>
<td>2: for ( i ) in range(( n_{\text{heads}} )) do</td>
</tr>
<tr>
<td>3: ( \text{attn}<em>{\text{append}}(\text{attn}</em>{\text{heads}}<a href="%5Ctext%7Binputs%7D">i</a>) )</td>
</tr>
<tr>
<td>4: end for</td>
</tr>
<tr>
<td>5: ( \text{concat}_{\text{attn}} \leftarrow \text{concat}(\text{attn}) )</td>
</tr>
<tr>
<td>6: ( \text{multi}<em>{\text{linear}} \leftarrow \text{linear}(\text{concat}</em>{\text{attn}}) )</td>
</tr>
<tr>
<td>7: end</td>
</tr>
</tbody>
</table>
3.4. Benchmark model development

This study compares the performance of the deep hybrid multi-head self-attention transformer model (TNET) to standalone Deep Neural Network (DL) models including Convolutional Neural Networks (CNNs), Long Short Term Memory Networks (LSTMs), Deep Neural Networks (DNNs), Gated Recurrent Units (GRUs), and Bi-Directional LSTMs (BILSTMs). The hyperparameters of these standalone models are also derived using HyperOpt, and they are shown in Table 3 where the batchsize and epochs are 10 and 2000, respectively for benchmark models.

Additionally, the TNET and benchmark models use the Adam optimisation technique to optimise the training phase. In DL, Adam optimisers are widely used due to their self-adaptive learning rates and good convergence speeds [82]. In the Adam method, the first and second moments of the gradient are calculated to dynamically modify the learning rate of each parameter. Therefore, the parameter update is stable and does not easily fall into the local optimum. Additionally, CNN, LSTM, DNN, GRU, and BILSTM layers use Rectified Linear Units (ReLUs). As compared to sigmoid and tanh functions, the ReLU significantly improved accuracy and convergence speed in DL models by applying identity mapping on the positive side and discarding the negative input, resulting in efficient gradient propagation at training [83]. During training, two keras callback application programming interfaces were also used.

• Early Stopping (es): The es is employed to overcome the overfitting issue. Early stopping mechanism is used to halt model training before the initial set of epochs has finished. Model parameters are regularly evaluated, and training is terminated early if they begin to deteriorate [84]. In this study, Root Mean Square Error (RMSE) on validation data is set as performance metrics and patience is used as 20 for es.

• ReduceLROnPlateau: The ReduceLROnPlateau callback, like the Early Stopping callback, monitors a specified model parameter, such as validation accuracy. Rather than halting training, it shrinks the optimiser learning rate to minimise bouncing around the minima, which is frequently produced by a high learning rate [85]. We restricted the learning rate to a lower bound of $1 \times 10^{-8}$ and set the patience parameter to 10 epochs, with a decrease factor of 0.2.

3.5. Model point prediction accuracy measures

The proposed TNET vs. the CNN, LSTM, DNN, GRU and BILSTM models applied in point-based daily $G$ prediction were measured via a number of deterministic metrics. The mathematical representation are provided in Tables 4(c), 4(a) and 4(b).
To evaluate the proposed TNET, and counterpart models, we present three distinct classes of performance indicators computed in the testing phase. In the Class A measure shown in Table 4(a), the indicators of dispersion (or “error”) of individual points are shown (0 for a perfect model). According to Li et al. [46,89], the model performance could be excellent if the relative error (i.e. $R_{RMSE}$ and $R_{MAE}$) are $\leq 10\%$ and good if relative errors are between 10% to 20%. The Class B measures (Table 4(b)) are overall performance indicators with a maximum value of 1 for a perfect model [90–92]. Finally, the Class C utilising $KSI$ and $OVER$ indicate the distribution similarity (i.e., a lower value indicates greater similarity). When comparing datasets, the $KSI$ measures the distance between the Cumulative Distribution Functions (CDFs), but the $OVER$ measure only the distance where the critical value distance is exceeded. Gueymard [93] suggested new performance index called Combined Performance Index ($CPI$), it is used to optimally combine $RMSE$, $KSI$ and $OVER$ into single statistical indicator.

Besides the class A, B and C metrics, this study have also employed Global Performance Indicator (GPI) [94] as a metric to rank the models. Further, the Promoting Percentages ($\lambda$), Directional Symmetry ($DS$), the Diebold–Mariano ($DM$) [95] and the Harvey–Leybourne–Newbold ($HLN$) test statistic is used to compare the performance of TNET model with benchmark models in point prediction of daily $G$. To compare the statistical significance between two models, we use $DM$ statistic designed to compare the null hypothesis of equal predicted accuracy against the alternative of varying prediction skills across models. The null hypothesis is that the $RMSE$ generated from the tested model is greater than or equal to that from the reference
Energy and AI 14 (2023) 100302

Table 4(a)

<table>
<thead>
<tr>
<th>Deterministic performance measure (Class A)</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation Coefficient</td>
<td>( r = \frac{\sum (G^o - \bar{G}) (G - \bar{G})}{\sqrt{\sum (G^o - \bar{G})^2 \sqrt{\sum (G - \bar{G})^2}}} ) (42)</td>
</tr>
<tr>
<td>Root Mean Square Error (MW)</td>
<td>( RMSE = \sqrt{\frac{1}{n} \sum (G^o - G)^2} ) (43)</td>
</tr>
<tr>
<td>Mean Absolute Error (MW)</td>
<td>( MAE = \frac{1}{n} \sum</td>
</tr>
<tr>
<td>Relative Root Mean Square percentage Error (%)</td>
<td>( RRMSPE = \frac{\text{RMSE}}{G^o} \times 100% ) (45)</td>
</tr>
<tr>
<td>Relative Mean Absolute Percentage Error (%)</td>
<td>( RMAPE = \frac{\text{MAE}}{G^o} \times 100% ) (46)</td>
</tr>
<tr>
<td>Uncertainty at 95%</td>
<td>( U_{95} = 1.96 \times SD ) (51)</td>
</tr>
<tr>
<td>1-statistic</td>
<td>( TS = \sqrt{\left( \frac{n-1}{n} \text{RMSE} \right)^2} ) (48)</td>
</tr>
<tr>
<td>Mean Bias Error (MW)</td>
<td>( MBE = \frac{(100/ (G^o))}{N} \sum (G^o_i - G_i) ) (49)</td>
</tr>
<tr>
<td>Standard deviation of the Relative Error</td>
<td>( STDRE = \left( \frac{1}{n} \sum (G^o_i - G_i)^2 \right)^{1/2} ) (50)</td>
</tr>
<tr>
<td>Explained Variance Score</td>
<td>( E_{var} = 1 - \frac{\text{VAR}^2}{\text{VAR}^2} ) (51)</td>
</tr>
<tr>
<td>Absolute Percentage Bias (%)</td>
<td>( ABP = \frac{\sum</td>
</tr>
<tr>
<td>Skill Score</td>
<td>( SS = 1 - \frac{\text{RMSE}^2}{\text{RMSE}(G^o)} ) (53)</td>
</tr>
</tbody>
</table>

Note: where \( G^o \) and \( G \) are the observed and predicted value of \( G \), \( G^o \) and \( G \) are the observed and predicted mean of \( G \), \( p \) stands for the model prediction, \( s \) for the observation, \( pr \) for perfect prediction (persistence), and \( r \) for the reference prediction, \( VAR \) is the variance, \( SD \) is the standard deviation, \( n \) corresponds to the size (number) of predictions [86].

Table 4(b)

<table>
<thead>
<tr>
<th>Deterministic performance measure (Class B)</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Willmott's Index</td>
<td>( E_{W1} = 1 - \frac{\sum (G^o - G)^2}{\sum (G^o - G)^2} ) (54)</td>
</tr>
<tr>
<td>Nash-Sutcliffe Equation</td>
<td>( E_{NS1} = 1 - \frac{\sum (G^o - G)^2}{\sum (G^o - G)^2} ) (55)</td>
</tr>
<tr>
<td>Legates and McCabe's Index</td>
<td>( E_{LM} = 1 - \frac{\sum (G^o - G)^2}{\sum (G^o - G)^2} ) (56)</td>
</tr>
<tr>
<td>Theil's Inequality</td>
<td>( TIC = \sqrt{\left( \frac{1}{n} \sum (G^o_i - G_i)^2 \right)^{-1} - \left( \frac{1}{n} \sum (G^o_i - G_i)^2 \right)^{-1}} ) (57)</td>
</tr>
<tr>
<td>Kling-Gupta Efficiency</td>
<td>( KG1 = 1 - \left( v - 1 \right)^2 + \left( \frac{v}{v - 1} \right)^2 ) (58)</td>
</tr>
</tbody>
</table>

Note: where \( G^o \) and \( G \) are the observed and predicted value of \( G \), \( G^o \) and \( G \) are the observed and predicted mean of \( G \), \( s \) is the number (number) of predictions, \( CV \) Coefficient of Variation.

Table 4(c)

<table>
<thead>
<tr>
<th>Deterministic performance measure (Class C)</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>KSI</td>
<td>( KSI = \frac{\delta}{\gamma} \frac{\int_{D_0}^{D_0} D_x d x}{\int_{D_0}^{D_0} D_x d x} ) (59)</td>
</tr>
<tr>
<td>Critical Limit Overestimation Index</td>
<td>( OVER = \frac{\delta}{\gamma} \frac{\int_{D_0}^{D_0} D_x d x}{\int_{D_0}^{D_0} D_x d x} ) (60)</td>
</tr>
<tr>
<td>where ( A_x = D_x(X_{sec},X_{sec}) )</td>
<td>(61)</td>
</tr>
<tr>
<td>and ( D_x = \Phi(N)^{1/2} )</td>
<td>(62)</td>
</tr>
<tr>
<td>Combined Performance index</td>
<td>( CPI = \frac{\text{KSI}}{\text{GPI}} \times \text{RMSE} ) (63)</td>
</tr>
</tbody>
</table>

Note: Where \( D_x \) is the absolute difference between the calculated and measured CDF, \( X_{sec} \) and \( X_{sec} \) are the minimum and maximum values of \( D_x \), \( A_x \) is the critical area, \( D_x \) is a statistical characteristic of the reference distribution or critical value, \( N \) is the number of points and \( \Phi(N) \) is a pure function of \( N \) [87,88].

model. The DM statistic is defined as:
\[
DM = \frac{\bar{g}}{\sqrt{V_g N}} \tag{64}
\]

Note: Where \( \bar{g} = \frac{\sum g_i}{N} \), \( g_i = \left( \sum s_i - s_{i,t} \right)^2 - \left( x_i - x_{i,t} \right)^2 \), and \( V_g = \gamma_0 + 2 \sum t_i \left( \gamma_i = \text{cov} \left( s_{i+1}, s_i \right) \right) + \gamma \), is the variance of \( g_i \), \( x_{i,t} \) and \( x_{i,t} \) signifies the predicted \( x \) computed using the test \( t \) and reference method \( t \), respectively, in period \( i \) while \( N \) = number of observations in hold-out sample.

Furthermore, \( GPI \) was calculated using the six different metrics as follows:
\[
P_{GI} = \frac{1}{n} \sum a_j (g_j - y_{ij}) \tag{65}
\]

where \( a_j \) = median of scaled values \( g_j \) of statistical indicator \( j \) for model \( i \), with \( j = 1 \) for \( RMSE, MAE, MAPE, RRMSPE, J = 1, 2, 3, 4, 5 \) for \( MBE \), and \( j = -1 \) for \( r \).

The Directional Symmetry (DS) calculates the degree of agreement between predicted and actual values in the direction of tendency.
(i.e., up or down), defined as follows:
\[
DS = \frac{1}{n} \sum_{i=2}^{n} d_i \times 100\%
\]  
(66)
where,
\[
d_i = \begin{cases} 
1 & \text{if } (G_i^m - G_{i-1}^m)(G_i^p - G_{i-1}^p) > 0 \\
0 & \text{otherwise} 
\end{cases}
\]  
(67)

The Promoting Percentage of Absolute Percentage Bias \( \lambda_{APB} \), Theil’s inequality coefficient \( \lambda_{TIC} \), and RMSE \( \lambda_{RMSE} \) are used to compare various models.

\[
\lambda_{APB} = \left| \frac{(APB_1 - APB_2)}{APB_1} \right|
\]  
(68)

\[
\lambda_{TIC} = \left| \frac{(TIC_1 - TIC_2)}{TIC_1} \right|
\]  
(69)

\[
\lambda_{RMSE} = \left| \frac{(RMSE_1 - RMSE_2)}{RMSE_1} \right|
\]  
(70)
where \( APB_1, RMSE_1 \) and \( TIC_1 \) are objective model performance metrics, and \( APB_2, RMSE_2 \) and \( TIC_2 \) are benchmark model performance.

3.6. Quantifying the uncertainty: Interval prediction

The Prediction Intervals (PIS) estimations in the second stage of the modelling were used to make interval-based predictions as opposed to a point prediction in the first stage. PI of \( G \) were quantified using nonparametric Kernel Density Estimation (NKDE) method at 95% confidence level with the following steps to generate the PISs.

- The probability distribution (PD) of prediction errors (PE) produced from TNET and benchmark models is first assessed. To produce the probability density function (PDF) to fit the PDs, the NKDE approach was used.
- To represent potential changes in \( G \) predictions, PIS were computed using the NKDE-PDF approach.
- The estimated PISs were evaluated by using probabilistic metrics (Table 4(d)).

In the sector, confidence intervals among any predictions are more influential than point-based predictions of energy demand. In order to assess predictive performance, we use the Mean Prediction Interval Width (MPIW) and Prediction Interval Coverage Probability (PICP). These prediction points fall in both cases within the interval and span the interval as seen in Table 4(d). Generally, a higher coverage probability (PICP) and a lower interval width (MPIW) are expected, and PICP should be near or greater than 0.95. Therefore, high-quality \( PI \) have relatively large PICP values and low MPIW values.

On the other hand, in order to assess the performance in respect to the prediction interval, a comprehensive index \( F \) is used. This index evaluates the performance of the prediction interval by obtaining large PICP values by widening the prediction intervals. In order to determine the best \( PI \), \( F \) combines both opposing indexes where a higher \( F \) is generally a better interval prediction model. The quality of \( PI \) is also measured using metrics like Prediction Interval Normalized Average Width (PINAW), Average Relative Interval Width (ARIL) and Winkler Score (WS). For a nominal confidence level, a PI with high quality has a lower value of the PINAW, ARIL, and WS. Finally, the Continuous Rank Probability Score compares the cumulative distribution functions (CDFs) of predicted and observed probability distributions. Generally, a lower CRPS indicates better probabilistic performance.

4. Results and discussion

4.1. Deterministic results

The performance of the proposed deep hybrid TNET model was assessed using statistical score metrics provided in Table 4(a), 4(b), and 4(c) as well as diagnostic charts. The standalone models (i.e., LSTM, BILSTM, GRU, CNN, and DNN) were compared with the TNET model in the testing period. The objective model (TNET) with the lowest values for RMSE, MAE, RMSPE, MAPE, SS, APB, and ST DRE and the highest values for KGE, ENs, R, ELM, EW, and Ew were selected and models were then ranked according to G1.

As for Bethania, the proposed TNET model employed for \( G \) prediction produced a high \( r \) value, as well as low MAE and RMSE values (\( r \approx 0.970, RMSE \approx 21.957 \), MAE \( \approx 17.056 \)). The CNN and LSTM models performed differently (\( r \approx 0.950, RMSE \approx 23.224, MAE \approx 18.176 \), respectively) when compared to CNN (\( r \approx 0.964, RMSE \approx 23.657, MAE \approx 18.415 \)). Similarly, the \( r \) value for other DL models (BILSTM, GRU, and DNN) is lower but the \( RMSE \) and \( MAE \) are both higher than those of the TNET model. The DNN model generated the third-top \( r \) value but also the worst values of \( RMSE \) and \( MAE \) (\( r \approx 0.951, RMSE \approx 30.193, MAE \approx 21.417 \)). Moreover, when benchmarking the remaining sites, the TNET yielded significantly better \( G \) predictions (Table 5 and Fig. 8). Fig. 8 depicts the scatter plots for observed and expected \( G \) in the testing phase. The proposed TNET, with its scatter points closer to the \( y = mx + c \) line, indicates that this model carries a viable option for \( G \) predictions.

Table 6 summarises the results of all DL models to predict daily \( G \) at four sub-stations using \( EW_1, EN_{NS}, \) and \( EL_{LM} \). Across study sites, the deep hybrid TNET model performed best, with a maximum magnitude of \( [|E_{W1}| / \text{approx} 0.924, |E_{NS}^1| / \text{approx} 0.829, \) and \( |E_{LM}^1| / \text{approx} 0.691 \) (e.g., Coopers Plains). These measures are in smaller magnitude for the other DL models such as the second-top GRU \( EW_1 \approx 0.914, EN_{NS} \approx 0.814, \) and \( EL_{LM} \approx 0.669 \) and the worst BILSTM model \( EW_1 \approx 0.903, EN_{NS} \approx 0.799, \) and \( EL_{LM} \approx 0.648 \). In addition, all predictive models produced consistently better values of \( EW_1, EN_{NS}, \) and \( EL_{LM} \) for Beenleigh and Bethania substations. However, this does not hold for the Browns Plains site where the BILSTM model produced the second-best magnitude of \( EW_1 \approx 0.865 \) but the GRU model was in second place in terms of \( EN_{NS} \approx 0.824 \) and \( EL_{LM} \approx 0.573 \). Aside from TNET, other DL models have varying predictive capacity across substations, except for the TNET model. For instance, the BILSTM model yielded the lowest values of \( EW_1, EN_{NS}, \) and \( EL_{LM} \) at the Coopers Plains sub-station but the second-best at the Beenleigh sub-station.

Further comparative accuracy (Table 7) revealed that the TNET model, as opposed to LSTM, BILSTM, GRU, CNN, and DNN, generated the lowest RMSPE and MAPE at all four study sites, other than a small variation at the Beenleigh site (Table 7). More specifically, at the Beenleigh site, the TNET model acquired the smallest RMSPE \( \approx 8.873 \% \) with MAPE \( \approx 6.560 \% \) compared to RMSPE \( \approx 8.935 \% \) and MAPE \( \approx 6.558 \% \) from the second-top BILSTM. Likewise, Table 8 showed that the TNET model produced the best performance in three out of four study sites (Coopers Plains, Beenleigh, and Bethania) based on the \( E_{w} \) score while at the Browns Plains site, the GRU model generated the highest value of \( E_{w} \approx 0.837 \) just above \( E_{w} \approx 0.836 \) from TNET. Furthermore, the TNET model shows the lower value of \( U95 \) uncertainty indicator compared to other benchmark models. For instance, the magnitude of \( U95 \) for TNET model is \( \approx 117.11 \) compared to \( \approx 125.93, \approx 126.85, \approx 122.06, \approx 124.61, \approx 125.11 \) for LSTM, BILSTM, GRU, CNN and DNN model respectively (Coopers Plains sub-station). It was also evident from the table that the \( TS \) metric was always low (\( TS < 2 \)) for the TNET model. For instance, in Browns Plains sub-station the \( TS \) for TNET model is \( \approx -1.445 \) compared to \( \approx 2.412, \approx 1.064, \approx 6.126, \approx 6.466, \approx 13.688 \) for LSTM, BILSTM, GRU, CNN and DNN model respectively. Similarly the Theil’s Inequality Coefficient (TIC) (Table 8) for the TNET model is lower than that of other comparative
Table 4(d)
Probabilistic performance measure (Class D).

<table>
<thead>
<tr>
<th>Deterministic performance measure (Class D)</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prediction Interval Coverage Probability</td>
<td>( P_{ICP} = \frac{1}{N} \sum_{i=1}^{N} e_i ) (71)</td>
</tr>
<tr>
<td>Mean Prediction Interval Width</td>
<td>( M_{PIW} = \frac{1}{N} \sum_{i=1}^{N} (U(G_i) - L(G_i)) ) (73)</td>
</tr>
<tr>
<td>F Value</td>
<td>( F = \frac{P_{ICP}}{\frac{P_{ICP} + E}{2}} ) (74)</td>
</tr>
<tr>
<td>Average Relative Interval Length</td>
<td>( AIRL = \frac{1}{N} \sum_{i=1}^{N} \frac{U(G_i) - L(G_i)}{G^\alpha} ) (75)</td>
</tr>
<tr>
<td>Winkler Score</td>
<td>( WS = \begin{cases} \frac{\Delta_i}{\alpha} &amp; (L(G_i) &lt; y) \ \frac{\Delta_i}{\alpha} + 2(L(G_i) - y)/\alpha &amp; (L(G_i) &lt; y &lt; U(G_i)) \ \frac{\Delta_i}{\alpha} &amp; (y &lt; L(G_i)) \ \frac{\Delta_i}{\alpha} + 2(y - U(G_i))/\alpha &amp; (y &gt; U(G_i)) \end{cases} ) (76)</td>
</tr>
<tr>
<td>Normalised Mean Prediction Interval Width</td>
<td>( P_{INAW} = \frac{1}{N} \sum_{i=1}^{N} \frac{(U(G_i) - L(G_i))}{\sum_{i=1}^{N} y_i} ) (78)</td>
</tr>
<tr>
<td>Continuous Rank</td>
<td>( CRPS = \frac{1}{N} \sum_{i=1}^{N} crps(F_i, y_i) ) (79)</td>
</tr>
</tbody>
</table>

Note: \( N \) denotes the number of test samples, \( y_i \) is the \( i \)th observation, \( L(G_i) \) and \( U(G_i) \) represent lower bound and upper bound of the \( i \)th. G Prediction Interval respectively, \( G^\alpha \) is the observed value of G, \( R \) is the Range. [97]. In CRPS metrics, \( 1(\cdot) \) is the Heaviside function, it takes the value of 1 when \( x > y \) and equals 0 otherwise.

Table 5
The testing performance of the Deep Hybrid Multi-Head Self Attention Transformer Model (TNET) vs. benchmark models as measured by Correlation Coefficient (\( r \)), Root Mean Square Error (RMSE, MW) and Mean Absolute Error (MAE, MW).

<table>
<thead>
<tr>
<th>Sub-station</th>
<th>Predictive model</th>
<th>Model performance metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( r )</td>
</tr>
<tr>
<td>Coopers Plains</td>
<td>TNET</td>
<td>0.965</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>0.948</td>
</tr>
<tr>
<td></td>
<td>BILSTM</td>
<td>0.946</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>0.950</td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>0.948</td>
</tr>
<tr>
<td></td>
<td>DNN</td>
<td>0.948</td>
</tr>
<tr>
<td>Browns Plains</td>
<td>TNET</td>
<td>0.956</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>0.953</td>
</tr>
<tr>
<td></td>
<td>BILSTM</td>
<td>0.951</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>0.958</td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>0.954</td>
</tr>
<tr>
<td></td>
<td>DNN</td>
<td>0.943</td>
</tr>
<tr>
<td>Beenhlev</td>
<td>TNET</td>
<td>0.942</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>0.935</td>
</tr>
<tr>
<td></td>
<td>BILSTM</td>
<td>0.941</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>0.912</td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>0.917</td>
</tr>
<tr>
<td></td>
<td>DNN</td>
<td>0.941</td>
</tr>
<tr>
<td>Bethania</td>
<td>TNET</td>
<td>0.970</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>0.950</td>
</tr>
<tr>
<td></td>
<td>BILSTM</td>
<td>0.931</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>0.843</td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>0.964</td>
</tr>
<tr>
<td></td>
<td>DNN</td>
<td>0.951</td>
</tr>
</tbody>
</table>

Table 6
The performance of the Deep Hybrid Multi-Head Self Attention Transformer Model (TNET) vs. LSTM, BILSTM, CNN, GRU and DNN models using the Willmott’s Index (\( EW \)), Nash–Sutcliffe Coefficient (\( NS \)) and the Legates & McCabe’s (\( LM \)) Index of Agreement. Note that the best model is boldfaced (blue).

<table>
<thead>
<tr>
<th>Sub-station</th>
<th>Predictive model</th>
<th>Model performance metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( \bar{E}_{\text{W}} )</td>
</tr>
<tr>
<td>Coopers Plains</td>
<td>TNET</td>
<td>0.924</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>0.905</td>
</tr>
<tr>
<td></td>
<td>BILSTM</td>
<td>0.903</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>0.914</td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>0.911</td>
</tr>
<tr>
<td></td>
<td>DNN</td>
<td>0.908</td>
</tr>
<tr>
<td>Browns Plains</td>
<td>TNET</td>
<td>0.874</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>0.858</td>
</tr>
<tr>
<td></td>
<td>BILSTM</td>
<td>0.865</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>0.848</td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>0.838</td>
</tr>
<tr>
<td></td>
<td>DNN</td>
<td>0.205</td>
</tr>
<tr>
<td>Beenhlev</td>
<td>TNET</td>
<td>0.903</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>0.893</td>
</tr>
<tr>
<td></td>
<td>BILSTM</td>
<td>0.902</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>0.854</td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>0.859</td>
</tr>
<tr>
<td></td>
<td>DNN</td>
<td>0.891</td>
</tr>
<tr>
<td>Bethania</td>
<td>TNET</td>
<td>0.907</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>0.901</td>
</tr>
<tr>
<td></td>
<td>BILSTM</td>
<td>0.850</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>0.894</td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>0.898</td>
</tr>
<tr>
<td></td>
<td>DNN</td>
<td>0.771</td>
</tr>
</tbody>
</table>

models. These analysis from Tables 7 and 8 further revealed that the deep hybrid TNET model capture the future \( G \) with higher accuracy.

Fig. 8 depicted the bar chart of \( SS \) to compare the \( G \) prediction capacity of the deep hybrid TNET and other benchmark DL models. It can be visualised that the TNET model yielded the highest \( SS \) value over the four study sites meanwhile the other model showed variable performance. It can also be seen that the \( SS \) of the LSTM model and
DNN model are less than 30% for the Bethania sub-station, the lower value in $SS$ is because the LSTM and DNN model are not performing well at Bethania sub-station but if we compare the same model (LSTM and DNN) at Coopers plains and Beenleigh sub-stations the $SS$ is more than 70%. In agreement with that, Figs. 9 and 10 also demonstrated that TNET is superior to other DL models in terms of the overall distribution of $|PE|$ and $ECDF$. In Fig. 9, it can be seen that the median, 75th percentile, upper whiskers, and possible outliers of $|PE|$
The comparison of the Deep Hybrid Multi-Head Self Attention Transformer Model (TNET) vs. other comparative models in terms of the relative errors (RMSE, %) and (MAPE, %) within the test site. Note that the best model is boldfaced (blue).

### Table 8

<table>
<thead>
<tr>
<th>Sub-station</th>
<th>Predictive model</th>
<th>Model performance metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>RMSE</td>
</tr>
<tr>
<td>Coopers Plains</td>
<td>TNET</td>
<td>10.265%</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>11.097%</td>
</tr>
<tr>
<td></td>
<td>BILSTM</td>
<td>11.109%</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>10.698%</td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>10.903%</td>
</tr>
<tr>
<td></td>
<td>DNN</td>
<td>10.967%</td>
</tr>
<tr>
<td>Browns Plains</td>
<td>TNET</td>
<td>5.936%</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>6.198%</td>
</tr>
<tr>
<td></td>
<td>BILSTM</td>
<td>6.293%</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>6.124%</td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>6.357%</td>
</tr>
<tr>
<td></td>
<td>DNN</td>
<td>9.550%</td>
</tr>
<tr>
<td>Beenleigh</td>
<td>TNET</td>
<td>8.879%</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>9.345%</td>
</tr>
<tr>
<td></td>
<td>BILSTM</td>
<td>8.935%</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>10.665%</td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>10.426%</td>
</tr>
<tr>
<td></td>
<td>DNN</td>
<td>9.377%</td>
</tr>
<tr>
<td>Bethania</td>
<td>TNET</td>
<td>6.864%</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>7.261%</td>
</tr>
<tr>
<td></td>
<td>BILSTM</td>
<td>9.293%</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>7.536%</td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>7.396%</td>
</tr>
<tr>
<td></td>
<td>DNN</td>
<td>9.439%</td>
</tr>
</tbody>
</table>

derived from the TNET model are lower than those from other DL approaches. Such values from the DNN model are visually the worst while the comparison between LSTM, BILSTM, GRU, and CNN is hard to conclude. Similarly, the visual analysis using the EDCF of PE (Fig. 10) clearly indicated the inferior performance of DNN and BILSTM at Browns Plains and Bethania sites but the distinction among the top models was generally small. This further confirms the suitability of the proposed deep hybrid Multi-Head Self Attention Transformer (TNET) predictive model in predicting daily $G$.

To gain a more thorough understanding of the proposed TNET model, Fig. 12 depicts the frequency distribution of the absolute prediction error ($|PE|$) utilising all models for four sub-stations. It is revealed from Fig. 12 that the proposed TNET model provide significantly improved distribution with 61% (Coopers Plains sun-station), 45% (Browns Plains sub-station), 60% (Beenleigh sub-station) and 65% (Bethania sub-station) $|PE|$ within first error brackets ($0 < |PE| < 25$), whereas Coopers Plains, $0 < |PE| < 30$: Browns Plains, $0 < |PE| < 20$: Beenleigh and Bethania). Interestingly, $G$ predictions at the Bethania site obtained by the deep hybrid TNET model had the highest frequency of errors within the lowest error band (20 MW), which included 65% of the test data. The BILSTM, GRU, DNN, LSTM, and CNN, on the other hand, accumulated 56%, 61%, 48%, 63%, and 63%, respectively (see Fig. 11).

Nevertheless, a substantial amount of predicted data with comparatively small absolute prediction errors demonstrated the proposed TNET predictive model’s considerable prediction skill. The proposed TNET model was assessed based on $KGE$ and $APB$ for all tested models to analyse the robustness of the proposed model further. According to Table 10, the $APB$ of the proposed TNET model is significantly low and $KGE$ is high for each station, which clarifies its merits. Despite the fact that different statistical indicators were used, no one statistic alone could be used to determine which model was the best at each sub-station.

Consequently, a global performance indicator ($GPI$) was applied to the present study. Fig. 13(b) shows more detail information about the performance of each model at four sub-station in terms of $GPI$, TNET model outperform all other comparative models with highest $GPI$. For instance, in Browns Plains sub-station, the $GPI$ for TNET is $≈ 0.144$ compared to $≈ 0. –0.018$, $≈ 0.017$, $≈ 0.048$, $≈ −0.054$ and $≈ −1.415$ for LSTM, BILSTM, GRU, CNN and DNN respectively. Therefore, the proposed deep hybrid Multi-Head Self Attention Transformer (TNET) predictive model exhibits excellent prediction performance than other models in daily $G$ prediction.

Using a Taylor diagram, this paper summarises the relationship between Standard Deviation, $RMSE$, and Correlation Coefficient intuitively to reflect the advantages of the proposed prediction model. Fig. 14 depicts the taylor diagram showing the objective model as well as the comparative models. As shown in the Taylor diagram, TNET has the closest relationship to observation, so it indicates that the predicted value of the TNET model fits better with the actual value, which proves that the deep hybrid TNET model is best. Furthermore, the analysis of the Taylor diagram can reveal whether there is a correlation between two sets of data, and the higher the correlation coefficient is, the greater the accuracy of the prediction.

Thus, the proposed prediction model TNET is the most effective. It is clear from these results that the proposed predictive model TNET is reliable and feasible. Additionally, the directional symmetry($DS$) for each prediction model is shown in Fig. 15. The $DS$ of the proposed TNET model offers advantages, it can be observed that TNET acquired the highest score of $DS ≈ 72%$ followed by LSTM and BILSTM with $DS ≈ 71%$ and GRU and CNN with $DS ≈ 70%$. The DNN exhibited the worst performance by $DS ≈ 68%$. Therefore, the prediction accuracy of the proposed prediction model (TNET) for daily electricity demand is better than the other comparative models.

Lastly, a Promoting Percentage is calculated in order to specifically describe the prediction results of different prediction models. Table 9 gives the result of the promoting percentage of $RMSE$ ($Δ_{RMSE}$), $KGE$($Δ_{KGE}$) and APB($Δ_{APB}$) from the table it was evident that the Promoting Percentage metrics is positive for all the comparative models, which proves that the proposed TNET model is superior to other comparative models.

Furthermore, Table 10 shows the result of the $DM$ and $HLN$ test result of all models. The $DM$ and $HLN$ test value between the
Fig. 9. Bar chart showing Skill Score Metric ($SS$) of the TNET model and all other comparable models.
Note: The persistence model considers that the $G$ at $t$ is equal to the $G$ at $t+1$. It assumes that the atmospheric conditions are stationary.

Fig. 10. Box plot exemplifying the veracity of the deep hybrid Multi-Head Self Attention Transformer (TNET) predictive model in terms of the overall distribution of the absolute value of the prediction error against five different predictive models. (a) Coppers plains, (b) Browns Plains, (c) Beenleigh, and (d) Bethania.

The proposed TNET model and the other prediction models is greater than $\text{Table 9}$, which means that the proposed TNET model performs significantly better at significance levels of 5%. Therefore, it can be concluded that the proposed deep hybrid Multi-Head Self Attention Transformer (TNET) predictive model has not only increased prediction accuracy, but also demonstrated a significant difference in terms of prediction.
accuracy level (Table 10), which further demonstrates the superiority of the proposed model.

4.2. Probabilistic results

In this section, we present and compare the results of PIs. Table 11 shows the \( \text{PICP} \), \( \text{MPIW} \), and \( F \) of PIs obtained by KDE for 95% confidence level. The PICP of the PIs of the objective model are almost same (99%) but when we examine the \( \text{MPIW} \), the TNET model have lowest \( \text{MPIW} \) compared to other comparative models.

Specifically, the values of \( \text{MPIW} \) are \( \approx 77.57 \), \( \approx 75.79 \), \( \approx 75.18 \), and \( \approx 33.01 \) for TNET model at Coopers Plains, Browns Plains, Beenleigh, and Bethania sub-station respectively. However, the corresponding result of \( \text{BILSTM} \) is \( \approx 87.39 \), \( \text{DNN} \) is \( \approx 99.52 \), \( \text{CNN} \) is \( \approx 75.82 \) and \( \text{GRU} \) is \( \approx 119.22 \) at Coopers Plains, Browns Plains, Beenleigh, and Bethania sub-station respectively. It should be noted that a higher value of \( \text{PICP} \) does not always imply better performance. In general, the narrower the gap between PICP and Prediction Interval Nominal Confidence level (95%), the higher the quality of the produced PIs. The proposed TNET model, on the other hand, has the narrowest width (i.e., the lowest value of \( \text{MPIW} \)). As a result, identifying the best model using merely the \( \text{PICP} \), and \( \text{MPIW} \) indices is difficult.

The additional metric \( F \) is used in this study where \( F \) is defined as the weighted harmonic average of \( \text{PICP} \) and \( 1/\text{MPE} \) to evaluate the quality of interval-based predictions. The \( F \) index results consistently indicated the better prediction of daily \( G \) of the deep hybrid TNET model. For example, the \( F \) at Bethania is \( \approx 1.79 \) for TNET model compared to \( \approx 1.76 \), \( \approx 1.74 \), \( \approx 1.73 \), and \( \approx 1.65 \) for \( \text{CNN} \), \( \text{LSTM} \), \( \text{DNN} \), \( \text{BILSTM} \), and \( \text{GRU} \) model respectively. Aside from the metrics \( \text{MPIW} \) and \( F \), the proposed KDE approach provides a noticeable improvement in the \( W S \) and \( ARI_{IL} \) metrics (Table 12). With the lowest value of both \( W S \) and \( F \) the TNET model has outperformed all the comparative models at all four sub-stations. This increased \( \text{MPIW} \), \( W S \) and \( ARI_{IL} \) value of the comparative models could be attributable to the fact that their inadequate deterministic prediction capabilities could exacerbate their probabilistic prediction performance.

Finally, the Continuous Rank Probability Score (CRPS), which takes both reliability and sharpness into account is used in this study for assessing the probabilistic prediction. For the comparison purpose,
Table 13 shows the PINAw and CRPS metrics for all models. It can be seen from the table that for Coopers Plains sub-station the PINAw metrics is almost the same but the CRPS varies. The TNET model have lowest CRPS: 19.76 compared to ≈ 44.44, ≈ 22.18, ≈ 20.75, ≈ 43.39, and ≈ 42.93 for LSTM, BiLSTM, CNN, DNN and GRU respectively. For all other stations the proposed deep hybrid Multi-Head Self Attention Transformer (TNET) predictive model yielded the lowest value of PINAw and CRPS compared to other comparative models which implies that the PIs constructed by the proposed method are more informative and competitive.

5. Conclusion

This paper presented a new approach for daily power use predictions using Multi-Head Self-Attention Transformer (TNET) model tested on Australian data. The proposed TNET model is shown to be highly effective in accurately predicting daily power use, and can be applied to other similar datasets. The model was trained to predict electricity demand (G, MW) based on local climate variables from Coopers Plains, Browns Plains, Beenleigh, and Bethania the four Energex sub-stations in South-east Queensland, which is Australia’s sunshine state. It has been proven that the TNET model is highly reliable and can be utilised in other contexts beyond predicting daily power use. Therefore, this model could greatly benefit energy systems in Australia and other countries with similar climates. In comparison with other deep learning models such as Long-Short Term Memory (LSTM), Bidirectional LSTM (BiLSTM), Gated Recurrent Unit (GRU), Convolutional Neural Networks (CNNs), and Deep Neural Networks (DNNs), the proposed TNET model produced the highest accuracy of predicted G. Moreover, the TNET model may be more beneficial than other deep learning models due to its superior accuracy. According to the findings and results of the study, the new power demand prediction mechanism has the following advantages over other predictive models.
Fig. 13a. Bar chart comparing the efficacy of the proposed deep hybrid TNET model in terms of the tested absolute percentage bias (\(APB,\%\)) and Kling–Gupta efficiency (\(KGE\)).

Fig. 13b. Global performance indicator (\(GPI\)) used to evaluate the proposed TNET model relative to seven other benchmarked models.
First and foremost, the TNET model is more stable than other models developed with single-head attention, since it can be trained with fewer hidden layers. Moreover, compared with a single-head attention model, TNET can gain information from different illustration sub-groups. A significant advantage of TNET over other prediction methods is also the ability to focus on a large dataset that has relatively affected the model’s precision. Furthermore, TNET can capture complex relationships within the dataset by using multi-head attention, making it capable of achieving higher accuracy than single-head attention models. By using TNET, the model can process multiple sub-groups at the same time and capture relationships between them. This enables it to identify correlations between different features that single-head attention models may not be able to detect, resulting in more accurate predictions. Additionally, due to its ability to process large datasets, TNET can gain insights from more data than other methods, improving its precision and accuracy. The self-attention of the TNET model also

Fig. 14. Performance assessment of the deep hybrid model TNET as well as benchmark models to daily prediction in form of Taylor diagrams during the testing phase.

Fig. 15. The performance of the proposed deep hybrid TNET model compared to other counterpart models under study in terms of directional symmetry (DS) criteria.
Table 10
Evaluation of the Deep Hybrid Multi-Head Self Attention Transformer Model (TNET) against comparison models in terms of: (a) The Diebold–Mariano (DM) test statistic, (b) The Harvey–Leybourne–Newbold (HLN) test statistic.

<table>
<thead>
<tr>
<th></th>
<th>TNET</th>
<th>LSTM</th>
<th>BILSTM</th>
<th>GRU</th>
<th>CNN</th>
<th>DNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>DM (a)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TNET</td>
<td>2.524</td>
<td>2.63</td>
<td>1.664</td>
<td>3.616</td>
<td>3.055</td>
<td>3.055</td>
</tr>
<tr>
<td>LSTM</td>
<td>0.109</td>
<td>-3.624</td>
<td>-0.803</td>
<td>-0.778</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BILSTM</td>
<td>-4.294</td>
<td>-1.002</td>
<td>-1.003</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRU</td>
<td>1.16</td>
<td>2.273</td>
<td>0.638</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNN</td>
<td>1.16</td>
<td>2.273</td>
<td>0.638</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HLN (b)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TNET</td>
<td>2.649</td>
<td>2.79</td>
<td>1.747</td>
<td>3.795</td>
<td>3.206</td>
<td>3.206</td>
</tr>
<tr>
<td>LSTM</td>
<td>0.115</td>
<td>-3.803</td>
<td>-0.843</td>
<td>-0.817</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BILSTM</td>
<td>-4.506</td>
<td>-1.051</td>
<td>-1.053</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRU</td>
<td>1.218</td>
<td>2.386</td>
<td>0.669</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: - The column of the table is compared with the rows, and if the result is positive, the model in the rows outperforms the one in the column; on the contrary, if it is negative, then the one in the column is superior. Note that the best model is boldfaced (blue).

Table 11
Probabilistic prediction results for 95% confidence level in terms of Prediction interval coverage probability (PICP), mean prediction interval width (MPIW), and F index for four substations. F index is defined as the weighted harmonic average of PICP and 1/MPIW and evaluates the quality of interval prediction. Note that the best model is boldfaced (blue).

<table>
<thead>
<tr>
<th>Sub-station</th>
<th>Pred. M.</th>
<th>PICP</th>
<th>MPIW</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coopers Plains</td>
<td>TNET</td>
<td>99%</td>
<td>77.57</td>
<td>1.68</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>99%</td>
<td>84.09</td>
<td>1.63</td>
</tr>
<tr>
<td></td>
<td>BILSTM</td>
<td>99%</td>
<td>87.99</td>
<td>1.65</td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>99%</td>
<td>78.83</td>
<td>1.60</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>99%</td>
<td>79.84</td>
<td>1.66</td>
</tr>
<tr>
<td></td>
<td>DNN</td>
<td>99%</td>
<td>83.95</td>
<td>1.56</td>
</tr>
<tr>
<td>Browns Plains</td>
<td>TNET</td>
<td>99%</td>
<td>75.79</td>
<td>1.78</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>99%</td>
<td>86.21</td>
<td>1.77</td>
</tr>
<tr>
<td></td>
<td>BILSTM</td>
<td>99%</td>
<td>89.72</td>
<td>1.76</td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>99%</td>
<td>90.44</td>
<td>1.75</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>99%</td>
<td>99.52</td>
<td>1.73</td>
</tr>
<tr>
<td></td>
<td>DNN</td>
<td>99%</td>
<td>84.59</td>
<td>1.77</td>
</tr>
<tr>
<td>Beenleigh</td>
<td>TNET</td>
<td>99%</td>
<td>57.18</td>
<td>1.66</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>99%</td>
<td>69.68</td>
<td>1.64</td>
</tr>
<tr>
<td></td>
<td>BILSTM</td>
<td>99%</td>
<td>64.78</td>
<td>1.62</td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>99%</td>
<td>75.82</td>
<td>1.65</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>99%</td>
<td>61.50</td>
<td>1.65</td>
</tr>
<tr>
<td>Bethania</td>
<td>TNET</td>
<td>99%</td>
<td>83.01</td>
<td>1.89</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>99%</td>
<td>70.30</td>
<td>1.92</td>
</tr>
<tr>
<td></td>
<td>BILSTM</td>
<td>99%</td>
<td>73.24</td>
<td>1.93</td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>99%</td>
<td>78.83</td>
<td>1.94</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>99%</td>
<td>68.38</td>
<td>1.95</td>
</tr>
</tbody>
</table>

Table 12
Probabilistic prediction results for 95% confidence level in terms of Winkler Score (WS) and Average Relative Interval Length (ARIO) for four sub-stations. Note that the best model is boldfaced (blue).

<table>
<thead>
<tr>
<th>Sub-station</th>
<th>Predictive model</th>
<th>Model performance metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WS</td>
<td>ARIO</td>
</tr>
<tr>
<td>Coopers Plains</td>
<td>TNET</td>
<td>93.06</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>96.39</td>
</tr>
<tr>
<td></td>
<td>BILSTM</td>
<td>101.48</td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>92.08</td>
</tr>
<tr>
<td></td>
<td>DNN</td>
<td>89.08</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>89.31</td>
</tr>
<tr>
<td>Browns Plains</td>
<td>TNET</td>
<td>83.23</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>115.22</td>
</tr>
<tr>
<td></td>
<td>BILSTM</td>
<td>115.53</td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>119.25</td>
</tr>
<tr>
<td></td>
<td>DNN</td>
<td>103.97</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>114.26</td>
</tr>
<tr>
<td>Beenleigh</td>
<td>TNET</td>
<td>91.11</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>97.22</td>
</tr>
<tr>
<td></td>
<td>BILSTM</td>
<td>98.93</td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>83.48</td>
</tr>
<tr>
<td></td>
<td>DNN</td>
<td>64.38</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>92.86</td>
</tr>
<tr>
<td>Bethania</td>
<td>TNET</td>
<td>85.01</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>43.48</td>
</tr>
<tr>
<td></td>
<td>BILSTM</td>
<td>54.67</td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>41.34</td>
</tr>
<tr>
<td></td>
<td>DNN</td>
<td>45.67</td>
</tr>
<tr>
<td></td>
<td>GRU</td>
<td>66.38</td>
</tr>
</tbody>
</table>

makes the model work more intelligently by interacting with the inputs and determining the most important variables.

In addition, interval-based prediction of power use using the KDE-Gaussian approach was carried out based on the point prediction
results, which can provide more comprehensive prediction information for electricity producers and consumers. In conclusion, this research work introduces a robust electricity demand prediction model for daily forecasting using Australian stations located in Queensland. These findings indicate that the TNET model is more powerful in predicting power energy data from climate sources than conventional benchmarked models, which have shown significant capabilities to be applied to building an efficient renewable energy platform.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data were acquired from ENERGEX. https://www.energex.com.au.

Acknowledgements

The authors thank the data providers, all reviewers and Editor for their thoughtful comments, suggestions and review process. Partial support of this work was through a project PID2020-115454GB-C21 of the Spanish Ministry of Science and Innovation (MICINN).

References


S. Ghimire et al.


[67] Zhang S, Tay Y, Yao L, Sun A. Next item recommendation with self-attention. In: Pro-