Convolutional and Recurrent Autoencoder (CRAE) for Smart Meter Data Compression

Scott Kwong Hin Yip†
syip0005@student.monash.edu

***Abstract*—The wide-spread deployment of smart meters and increased demands for high-frequency energy data has yielded significant pressure on transmission and storage capabilities. To alleviate these pressures, a plethora of compression techniques have been proposed for energy data but no standard has been adopted thus far. In this paper, a deep neural network that amalgamates both long short-term memory (LSTM) and convolutional (CNN) layers is proposed. In comparison with previous deep learning based models, the inclusion of the LSTM layer enhances model expressiveness in learning long-term dependencies in addition to local dependencies. Furthermore, a generalised divisive normalisation (GDN) was applied to replace ReLU as the non-linear activation function to introduce a more adaptive feature extraction tool. In addition, the encoded representations were assessed on a downstream classification task. In simulations with three different datasets, the model achieved a modest enhancement in reconstruction error reduction compared to previous purely convolutional based models. However, the evaluations showed that discrete wavelet transform (DWT) consistently achieved the lowest reconstruction error performance. Finally, the encoded representations from the hybrid CRAE model achieved lower accuracy than pure convolutional models.**

***Index Terms*—Convolutional autoencoder, recurrent autoencoder, smart meter, deep learning, compression.**

I. Introduction

SMART meters, as part of advanced metering infrastructure (AMI), have been deployed over the last two decades to record and analyse customers' power consumption behaviour. In just the US alone there are 102.9 million smart meters [1] and the Victorian Government has mandated pervasive roll-out of smart meters since 2006 [2]. AMI is used to describe the system which incorporates both smart meters, management systems and communication networks [3], and allows for two-way communication of smart grid data.

Smart meters record high-resolution electrical energy data (EED) which has led to a shift from previous billing supply-side to demand-side management in energy systems. AMIs, using the data from smart meters, are thus used to address the challenges of energy systems, improve policies and improve quality of living [4]. For example, energy operators actively use EED to perform: probabilistic forecasting of loads to aid in grid management [5]; non-intrusive load management (NILM) to decompose power signals into appliance signals, with wider implications for energy resolution; and demand-side management for determining energy prices.

† S. Yip is a student of the Department of Data Science and AI, Monash University, Melbourne, Victoria, Australia.

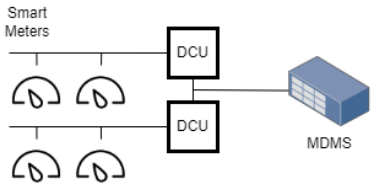

Fig. 1. Diagram of AMI communication structure.

A. Motivation and Related Works

EED is beginning to pose big data challenges, specifically volume and velocity. A smart meter which accrues voltage and current data at 16-bit precision with a frequency of 40 kHz will log data at approximately 39 kB/s [6]. In extrapolating to the 102.9 million smart meters in the U.S., this will result in approximately 2.1. million terabytes of EED generated per year. Furthermore, more recent data products require multi-source data beyond the limits of simply voltage, current and power. For example, there is demand for data related to temperature, barometric pressure and from renewable energy sources. Thus, as the volume and velocity of generated data reaches this magnitude, energy operators may experience network outages as limited bandwidth power line communication (PLC) reach capacity [7] and increasing data storage costs as the volume expands rapidly.

The big data issues also impact the choice of sampling rate for power companies. In order to avoid to circumvent such issues, power companies choose to limit their sampling rate to allay network demands [8]. The self-imposed limitation can reduce the effectiveness of data analytics tools which benefit from higher frequencies [9].

It must be clarified that these bottlenecks can occur in two places. As shown in Fig.1, smart meters communicate data to data concentrator units (DCU). DCUs then aggregate the data from meters to send to meter data management systems (MDMS) [10]. Bottlenecks can occur either between smart meters and DCUS, or DCUs and the MDMS. As such, any practical solution must consciously consider which bottleneck to relieve.

Data compression techniques are proposed as the pre-eminent solution to alleviate the physical and economic stresses.

Data compression is a well-researched topic in information theory, especially in the image and video domain. While many extant compression techniques have been developed for time-series data, the current international standard for smart meter

data communication IEC 62056-21 [11] does not specify any standardised choice of compression techniques.

Data compression is taxonomically divided into two subsets; lossless and lossy [12]. Lossless compression techniques encode and reconstruct compressed data with no error, while lossy compression techniques suffer source data loss. Reciprocally, lossy techniques compress source volume at higher ratios than lossless techniques and thus lossy techniques are the primary area of research for compression. Applications of lossy techniques are widespread such as in JPEG for images [13] and MP3 for audio [14]. It must be noted that lossless and lossy techniques do not work in isolation but can be used in tandem as an end-to-end technique [15].

The crux of lossy compression techniques is to map the source data into either a sparse representation, or a lower dimensional representation. Values which are close to zero are discarded [13] through a process called quantization. Typically, quantization is not performed directly on the source data but on a projection of the data. The difficulty with EED compression particularly is identifying a transformation technique which applies to a diverse range of energy consumption patterns.

The effectiveness of lossy techniques is measured by two criteria: compression ratio (CR) and reconstruction error. Compression ratio measures the ability of a technique to reduce source volume while reconstruction error measures the amount of source information loss.

Many ubiquitous lossy compression algorithms have been tailored for EED application. Singular value decomposition (SVD) and the fraternal principal component analysis (PCA) [9] have been explored. Julio Cesar Stacchini de Souza, et al. [16] proposed the application of SVD and contended that SVD outperformed DWT across all CR, and Rachit Mehra, et al. [17] extend these results by demonstrating that iterative-PCA serve to denoise the data. Wavelet-based transforms such as discrete Fourier transform (DFT) [18] and discrete wavelet transformations (DWT) [19] have both been applied for EED compression. An application of non-negative k-SVD is explored by Y. Wang, et al. [8] who demonstrated that EED load profiles could be decomposed in constituent partial usage patterns that yield superior reconstruction error results when compared to PCA and DWT across all CR. Lulu Wen, et al. [12] present a detailed description and qualitative comparison of these techniques.

The key limitations of these extant techniques pertain to a lack of transformation expressiveness, and scalability. In terms of transformation expressiveness, SVD and PCA are capable only of linear decompositions of the data matrix and thus are not capable of learning non-linear complexities that occur in day-to-day or hour-to-hour energy consumption. The DFT discards location information fully to retain frequency information and thus is not suitable for representing transient EED usage patterns. The DWT projection, while it does retain location and frequency information, learns coefficients specific to the explicit orthonormal basis function selected. However, different energy usage patterns may be best served by basis functions which are not yet discovered.

Deep neural networks (DNN) are applied to many smart meter analytics problems as a feature extraction tool, ranging from load forecasting to theft detection [5]. It is only natural that DNNs have been proposed for EED compression through the autoencoder (AE) architecture [20], whereby the latent vector is treated as the compressed representation of the EED. In comparison to many other lossy compression algorithms, AEs can learn the nonlinear function in projecting the input to its latent representation, thus allowing more complex relationships to be captured with fewer inductive biases. The result is lower reconstruction error, while maintaining a higher compression ratio.

Current state-of-the-art DNN-based EED compression algorithms are based on convolutional neural networks (CNN) [21]. The convolution operation can exploit local spatial and temporal localities of the energy data while retaining equivariance to shifts of locations in the feature map. H. Wang, et al. [22] illustrate through Markov-chains that energy consumption is locally dependent. For example, the authors illustrate for their dataset that residents generally do laundry and dishwashing prior to sleeping. Therefore, CNNs can effectively leverage these types of local dependencies to achieve better reconstruction results.

However, convolutions are only able to capture local dependencies and therefore long-term dependencies are not explicitly accounted for. For example, if the data is sampled at 15 minute intervals, the local dependencies identified by H. Wang, et al. [22] apply. However, relatively long-term dependencies, such as between every 12 hours, are not captured. It is natural to envisage and extend the previous example, that if dishwashing is done in the morning, then it may not be done at night. Existing research has not yet addressed this shortcoming for EED compression.

B. Contributions

Recurrent neural networks (RNN) are a category of DNN which is applied to sequential data. Certain variations such as the long short-term memory (LSTM) or gated recurrent unit (GRU) are able learn long-term dependencies. RNNs are already applied to many EED tasks such as anomalous energy detection [23], probabilistic load forecasting [24], [25] and NILM [26]. Furthermore, CNN-RNN EED hybrid models have also been explored to great effect for theft detection [23], [27], load forecasting [28], [29] and non-intrusive load monitoring [30], [31].

This paper proposes and examines a hybrid CNN and RNN autoencoder (CRAE) for EED compression that aims to capture both local and long-term dependencies, and thus seek to achieve state-of-the-art compression. For the encoder, we insert both depth-wise separable convolutional filters and LSTM cells in the encoder while ensuring that the architecture remains lightweight. For the decoder, we retain the transposed convolution operations from Shouxiang Wang, et al. [21] to reconstruct the source data. We acknowledge that the use of an LSTM layer will increase the number of parameters for any reasonable hidden layer size, but we seek to constrict memory and CPU usage such that CRAE can be deployed on a smart meter.

The main contributions of this paper are summarised as follows:

1) We design a lightweight hybrid autoencoder architecture that exploits that the local and long-term characteristics of EED. In comparison to pure convolutional-based architectures such as SCSAE [21], we find that reconstruction error is modestly reduced. The number of parameters has increased for lower frequency data but is still lightweight enough to encode on computationally scarce smart meters. However, the results show that alternative compression techniques, such as Discrete Wavelet Transform (DWT), yield lower reconstruction error than AE-based models and thus challenges the applicability AE-based models for EED compression.
2) We study the effects of three EED frequencies on CRAE's reconstruction error. We are unable to draw a conclusion as the effect of other factors, such as geographic, and model design, have a greater impact than the frequency itself.
3) We explore and compare the use of the latent representation from the proposed methodology on a downstream task against previous DNN models. We identify that the evaluation metrics from CRAE's latent representations are inferior to previous state-of-the-art AE-based models.

The remainder of this paper is organised as follows: Section II explores the attributes of DNN techniques, and metrics for reconstruction error and compression; Section III describes the structure of CRAE; Section IV describes the case studies and downstream task; Section V details and analyses the comparative performance of CRAE against benchmark compression techniques and the performance of the downstream task; Section VI concludes and presents pathways for future work.

II. Background

A. Principles of Autoencoders for Compression

The AE is an unsupervised learning method based on DNNs. The basic AE is comprised of linear layers followed by a non-linear activation function. We note $a^{[l]}$ to be the activation of layer l then a basic block can be described as:

$$\mathbf{a}^{[l+1]} = f(\mathbf{W}^{[l]}\mathbf{a}^{[l]} + \mathbf{b}^{[l]}) \tag{1}$$

Where $f(\cdot)$ is the non-linear activation function, $\mathbf{W}$ are the weights and $\mathbf{b}$ is the bias of layer l. If we denote L to be the number of layers then the first activation $a^{[1]}$ is the input x, and $a^{[L]}$ is the reconstructed input $\hat{x}$. AEs are comprised of an encoder and a decoder, and generally follower a bottleneck structure as shown in Fig.2. The encoder performs an analysis transform of the input into its latent representation, and the decoder performs a synthesis transform to reconstruct an approximation $\hat{x}$ from that latent representation. The AE is generally trained with a mean squared error (MSE) function and as such the optimization problem is formulated as follows:

$$\mathbf{W}*, \mathbf{b}* = \underset{\mathbf{W}*,\mathbf{b}*}{\operatorname{argmin}} \left[\frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{x}_i)^2 \right] \tag{2}$$

where n denotes the total number of samples.

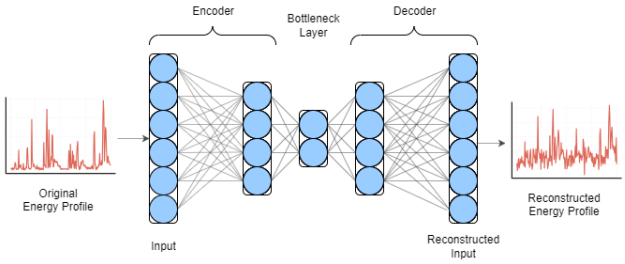

Fig. 2. Typical AE structure.

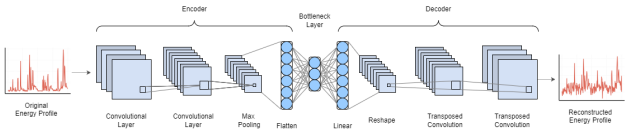

Fig. 3. Typical convolution AE structure.

To the best of our knowledge, Xiaoyao Huang, et al. [32] first proposed the use of AEs for EED compression. The authors also illustrated that AEs, trained with a greedy layer-wise wake-sleep method, comprehensively outperform DWT, k-SVD and PCA for all CRs. However, the authors only demonstrated this on an abated dataset comprised of representative load profiles only, thereby minimising granularity and noise. Jihoon Lee, et al. [33] proposed a holistic frequency-based framework which utilised the AE from Xiaoyao Huang, et al. [32] as the base. The authors identified that high frequency EED signals performed poorly with AEs, and thus proposed that separate AEs be trained on the dataset partitioned into high and low frequency. The holistic framework generated reconstruction error results which are superior to the original AE.

More recently, state-of-the-art reconstruction results across all CR have been obtained by using convolutional autoencoders (CAE) [21], [34], [35].

The convolutional layer was originally inspired by the receptive fields in the visual cortex and has been popularised in recent years for its image applications. Specifically, the convolution operation performs a linear operation between a sliding filter and the input. For a two-dimensional input, the convolutional layer can be defined as:

$$\mathbf{a}^{[l],f}(i,j) = f([\sum_{c \in C} \sum_{\delta_i=0}^{F-1} \sum_{\delta_j=0}^{F-1} \mathbf{a}^{[l-1],c}(i-\delta_i, j-\delta_j) \mathbf{W}^{[l],f}(i,j)] + b^{[l],f}) \tag{3}$$

where $a^{[l],f}(i,j)$ is the activation of the channel f of the layer l at row i and column j, $f(\cdot)$ is the non linear activation function, C refers to the total number of channels of the previous layer $[l-1]$, $W^{[l],f}$ refers to the weights of filter f on layer l and $b^{[l],f}$ refers to the bias for channel f on layer l.

The transposed convolution is a fractionally strided convolution which is used effectively as an inverse convolution. While strictly speaking it is not an inverse convolution [36], the behaviours of the transposed convolution replicate those

of a deconvolution in effect. As such, for a two-dimensional input, the convolution operation is commonly used to extract features into an activation of reduced width and length, but increased channels. Contrarily, the transposed convolution is used to expand the reconstruct the activation into an expanded width and length, but with reduced channels. Thus, for AEs, convolutional layers are integrated into the encoder and transposed convolution layers are integrated in the decoder. Fig.3 shows the general structure of the convolution AE as employed in [21], [34], [35].

As mentioned previously, CAEs exploit temporal-spatial dependencies in the data. This can be done by structuring the input EED as a two-dimensional image. Yuxuan Yuan, et al. [35] demonstrate that CAEs yield better reconstruction performance for voltage and power data compared to PCA and DWT. Shouxiang Wang, et al. [21] astutely note that CAEs require fewer parameters than its respective AEs. Furthermore, they suggested further parameter reduction by replacing the convolution operation with the lighter depth-wise separable convolution [37]. The authors demonstrate that this lighter CAE architecture still generates superior results when compared to k-SVD, PCA and AEs across all investigated CRs. In addition, S. Ryu, et al. [34] investigated using yearly load profiles as input data and purported that the CAE bore lower reconstruction error when compared exhaustively against PCA, kernel PCA, AE and DWT. Therefore, the authors demonstrate that larger input yearly load profiles, in addition to smaller daily load profiles, can be compressed effectively.

Recurrent neural networks (RNN) are a category DNN which processes inputs in a sequential manner. Certain variations such as the gated recurrent unit (GRU) and long-short-term memory (LSTM) can learn global dependencies in the input sequence. Recurrent units and its variants are commonplace in feature extraction for EED downstream tasks such as probability forecasting [25], energy theft detection [23] and NILM [26]. The use of RAEs for compression has been studied in other domains such as for images [38], and electrocardiogram signals [39].

The LSTM cell extends the RNN through the inclusion of memory cell in addition to the hidden state. RNNs suffer from a vanishing and exploding gradient problem if the input sequence length is too long, and the memory cell assists the network in overcoming this by allowing the network to learn the identity function where required. The cell state is determined by three gate units: the input, output and forget gates. Specifically, the input gate controls the flow from the input to the memory cell, the output gate controls the flow from the input to the hidden state and the forget gate limits the memory cell information carried forward. We denote the input gate as i_t, forget gate as f_t, output gate as O_t, candidate memory cell as $\tilde{c}_t$, memory cell as c_t and the hidden state as h_t. The mathematical formulations are expressed below:

$$i_t = f(x_t U^i + h_{t-1} W^i) \tag{4}$$

$$f_t = f(x_t U^f + h_{t-1} W^f) \tag{5}$$

$$o_t = f(x_t U^o + h_{t-1} W^o) \tag{6}$$

$$\tilde{C}_t = \tanh\left(x_t U^g + h_{t-1} W^g\right) \tag{7}$$

$$C_t = f(f_t * C_{t-1} + i_t * \tilde{C}_t) \tag{8}$$

$$h_t = \tanh\left(C_t\right) * o_t \tag{9}$$

CNN and RNN hybrid networks have been implemented with the intention of using CNNs to learn local dependencies and RNN's to learn global dependencies. These hybrid networks have been applied to human activity recognition [40], language modelling [41] and electroencephalogram classification [42], and sentiment analysis [43]. The motivation for these authors is commonly to exploit both window and temporal data. For the above four papers, the authors compared their hybrid models against pure RNN-based or pure CNN-based models and found that the hybrid model outperformed the solitary architectures which yields support to their motivations. Furthermore, for these papers, the adopted practice is to place that the convolutional operation occurs first, which is then fed into the recurrent cell thereby reducing the total number of parameters compared to the reverse sequence.

B. Evaluation Criteria

1) Measure of Volume Compression: The commonly deployed metric for measuring a technique's ability to reduce source volume is compression ratio (CR), as defined in (10).

$$\text{CR} = \frac{\text{Size of uncompressed data}}{\text{Size of compressed data}} \tag{10}$$

Strictly speaking, compression ratios evaluate file sizes. Instead, this paper utilises the encoding ratio (ER) metric as proposed by Shouxiang Wang, et al. [21]. ER is defined as the ratio of sampling points before and after sparse coding and shown in (11).

$$\text{ER} = \frac{\textit{raw sampling rate}\text{: } n \text{ raw data points per input}}{\textit{sampling rate post}\text{: } m \text{ data points per input}} \tag{11}$$

CR is influenced by a variety of factors, including choice of dataset [20], quantization function [13], and file systems . Consequently, CR varies significantly based on factors external to the compression algorithm singularly. Conversely, ER permits a fair comparison of volume compression between compression techniques extrinsic to the aforementioned details.

2) Measure of Reconstruction Error: The most widely employed reconstruction error metrics are MSE and mean absolute error (MAE) are expressed in (12) and (13) below, where x_t is the original data at time t, $\hat{x}_t$ is the reconstructed data at time t , and the number of data points is N.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{n}(x_i - \hat{x})^2 \tag{12}$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{n}|x_i - \hat{x}| \tag{13}$$

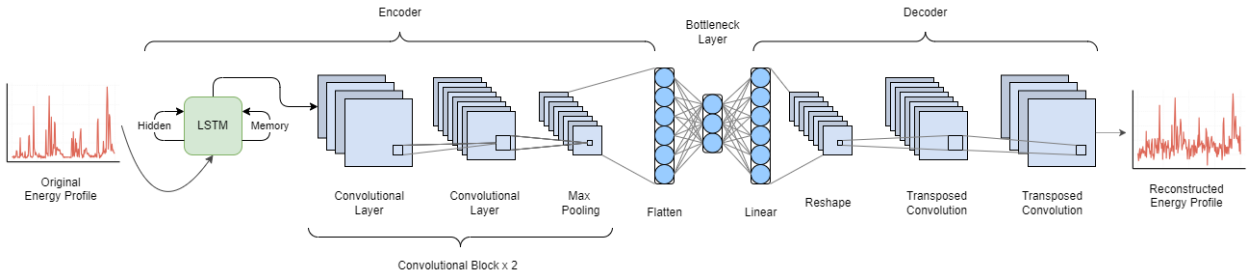

Fig. 4. CRAE architecture structure.

As all AE-based models will be trained using MSE, for convenience, this paper also utilises MSE as its reconstruction error evaluation metric.

III. Proposed Method

A. Design Principles

These design principles were briefly introduced in the previous sections. The core principles and the requirements that follow are summarised: (1) Low reconstruction error. The collected data will be used as the input into the data systems of energy operators. These can be user-sensitive systems such as billing, or energy theft detection. Furthermore, the higher parameter RNN must justify its additional computational cost. (2) Satisfices computational constraints for smart meters. As the intended application of CRAE is at the smart meter level, the encoder must be lightweight enough to run. However, it must be noted that this does not mean as lightweight as possible, rather simply meaning that it satisfices memory and CPU requirements. (3) Assumption of large datasets. While our case studies in Section IV may not be big data sets, the model must be designed with the anticipation that an energy operator will deploy this to all their customers. Thus, the model must be able to capture complexities from a complex distribution.

B. Detailed CRAE Network Structure

We draw inspiration and use the latest state-of-the-art model proposed SCSAE by Shouxiang Wang, et al. [21] as the foundation of CRAE. As such, we ensure that the CRAE remains comparable to SCSAE for both performance and more detailed analysis by holding many hyperparameters constant. Specifically, we retain the use of the depth-wise separable convolution in the encoder to maintain lightweightedness. The two key changes are as follows: the replacement of the first convolutional layer with an LSTM layer; and the use of the generalized normalization transformation (GDN) [15], [44] as the convolutional block activation function and batch normalisation simultaneously. The CRAE architecture is shown in Fig.4. Finally, we detail implementation details.

For the recurrent module, the alternatives were LSTM, GRU or the vanilla RNN. Vanilla RNN is dismissed due to its intrinsic issues with vanishing and exploding gradients. The choice between LSTM and GRU was more intricate as both were developed in response to the vanishing gradient issue and possess memory capabilities. Upon referring to extant literature, GRU appears to consistent perform better on smaller datasets and trains in less time than LSTMs in both time series forecasting [45] and text-based sentiment classification [46]. Contrarily, LSTMs perform significantly better on larger datasets where GRUs suffer serious performance degradation. This has been shown for sentiment classification [47], and language modelling [48], [49]. As such, given the assumption of large datasets, we anticipate that the choice of LSTM will yield better reconstruction results while satisficing computational restrictions. Furthermore, the above was corroborated by conducting several informal experiments that illustrated LSTM's superior performance. The results are not shown due to page limitations.

Regarding the activation function, we draw inspiration from the image neural network compression domain where neural networks outperform existing methods. Johannes Ball, et al. [44] first proposed the GDN and have adopted it in their pre-eminent image compression papers [15], [50]. The GDN activation layer serves as an enhanced replacement for both the batch normalization and non-linear activation function. The authors showed that GDN was highly efficient in Gaussianising the statistics of natural images, more so than any pointwise nonlinearities such as ReLU or Sigmoid. In fact, GDN can be summarised as a parametric multivariate generalisation of the Sigmoid function. While our energy images are not natural images, parallels can be drawn between patches of a natural image and patches of energy data and the local dependencies that exist. Two examples of 30 by 30 uni-channel patches are visualised in Fig.5. DRED is an energy image while Kodak is a natural image. A similar transitional property can be seen in both images as well as clustering of colours. The natural image does appear to have more noise but from our informal experiments the GDN appears to be able to exploit the other similarities effectively.

The comprehensive structure are shown in Table I. For both structures, w is denoted as the width frequency, i.e., the daily frequencies of 48 and 96 for CER and UMASS, and minute frequency of DRED being 60. The Table I details the CRAE architecture for an input height that is divisible by 4. This is the input shape used in [21]. An additional structure is shown in the Appendix for number of days which satisfy number of days $\bmod 4 = 3$. The relevant changes occur in the decoder structure and as such does not impact our encoder hyperparameters. The purpose of this additional structure is

TABLE I
DETAILED NETWORK ARCHITECTURE OF CRAE.

Type	Layer	Parameter	Output shape
Encoder	Input layer	-	$(1, 12, w)$
	Reshape	-	$(12, w)$
	LSTM layer	hidden size: 24, number of layers: 2, activation: tanh	$(12, 24)$
	Reshape	-	$(1, 12, 24)$
	2D convolutional layer	filter number: 32, activation function: GDN, stride: $(2, 2)$, kernel size: $(3, 3)$, padding: 1	$(32, 6, 12)$
	2D separable convolutional layer	filter number: 64, activation function: None, stride: $(2, 2)$, kernel size: $(3, 3)$, padding: 1	$(64, 3, 6)$
	MaxPooling2D	kernel size: $(1, 2)$, stride: $(1, 2)$, activation function: GDN	$(64, 3, 3)$
	Flatten	-	576
	Linear layer	latent dimension: d, activation function: None	d
Decoder	Linear layer	latent dimension: $64 \times 3 \times \frac{w}{4}$, activation function: None	$64 \times 3 \times \frac{w}{4}$
	Reshape	-	$(64, 3, \frac{w}{4})$
	2D transposed convolutional layer	filter number: 64, activation function: GDN, stride: $(2, 2)$, kernel size: $(3, 3)$, padding: $(1, 1)$, output padding $(1, 1)$	$(64, 6, \frac{w}{2})$
	2D transposed convolutional layer	filter number: 32, activation function: GDN, stride: $(2, 2)$, kernel size: $(3, 3)$, padding: $(1, 1)$, output padding $(1, 1)$	$(32, 12, w)$
	2D convolutional layer	filter number: 1, activation function: sigmoid, stride: $(1, 1)$, kernel size: $(1, 1)$, padding: 0	$(1, 12, w)$

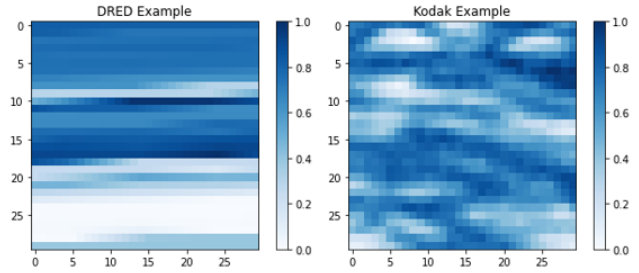

Fig. 5. 30×30 patch from energy image (left) and natural image (right).

TABLE II
COMPRESSION DATASET COMPARISON.

	Attributes				
Dataset	Sampling Rating	Location	Type	Customer #	Ref
CER	30 minutes	Ireland	Residential, Business	9181	[54]
UMASS	15 minutes	USA	Residential	114	[55]
DRED	1 second	Netherlands	Residential	1	[56]

to explore the effect of input days on reconstruction error in Section V-B.

Additionally, the AEs in this paper are implemented in PyTorch 1.12.1 in Python 3.7.0. All AEs in this paper are trained using AdamW [51] which is informally a weight decay corrected version of Adam. The default PyTorch AdamW hyperparameters are used with a weight decay set to 1e-10 set to prevent overfitting. The AEs were found to converge prior to 1000 epochs and thus all AEs were trained with 1000 epochs with the best test set loss being recorded. Batch size was set to 512 which was found to be an optimal balance between memory usage and convergence speed. No learning rate scheduler is applied, though experiments were performed with OneCycleLR [52] which did not have any effect on training or test loss.

IV. CASE STUDIES

A. Compression Datasets

Three public datasets with three different sampling rates are used as case studies. The information is summarised in Table II. The power units recorded in these datasets are real power measured in kilowatt hours.

The purpose of utilising three different datasets is to study the effects of sampling rate on CRAE and comparative techniques while retaining largely the same hyperparameters. A similar study was performed for lossless techniques by Andreas Unterweger, et al. [53]. To our knowledge, no detailed comparison of frequency on lossy techniques have been performed. However, we acknowledge that the records in the three datasets are generated from different distributions and thus comparative sampling rate observations must be cognisant of differences in geography, customer type and time periods.

B. Dataset Pre-processing

1) Linear Interpolation and Reshaping: The datasets above are real-life datasets and as such suffer from data loss or corruption due to sensor malfunction. As such, not only were there null values identified but there were also sequences of identical values identified through a prolonged period, indicating sensor malfunction. Each dataset was processed on customer-by-customer basis and each dataset was subject to different thresholds of acceptability. If the number of erroneous values fell below a threshold, then linear interpolation using forward fill was utilised. Alternatively, should the number of erroneous values exceed the threshold, list-wise deletion was performed for the entire daily block (CER and UMASS) or the entire hourly block (DRED).

To normalise seasonal effects, all the available data between the dates of 1 September to 15 October of the available year of each dataset. The number of customers for CER and UMASS

TABLE III
COMPARATIVE TECHNIQUE SETTINGS

	Models					
Attributes	PCA	FFT (1D)	DWT (1D)	DWT (2D)	k-SVD	AE
Sparse representation	Components	Non-zero coefficients	Non-zero coefficients	Non-zero coefficients	Components	Latent representation
Values	From 2 to 36	From 14 to 2880	From 14 to 2880	From 14 to 2880	From 1 to 5	From 14 to 2880
Other notes	solver: LAPACK SVD	-	wavelet: db2, decomposition level: 3	wavelet: db2, decomposition level: 3	max iteration: 60, transform algorithm: batch OMP	
References	[17]	-	[19]	[19]	[60]	[32]

were controlled to ensure that the size of all datasets was proportional.

The training and testing splits for CER and UMASS were split between customers to hold seasonality effects constant. Uniquely DRED has only one residential household, and thus the training and testing split was performed based on date periods instead. The resulting training dataset records prior to reshaping are $5184000, 518400, 5184000$ for CER, UMASS and DRED respectively. Similarly, the resulting dataset records prior to reshaping are $576000, 57600, 576000$ for CER, UMASS and DRED respectively.

Depending on the requirement of the input, the data was then, on a customer-by-customer basis, reshaped into the energy image into $\mathbb{R}^{\text{number of rows} \times \text{wide frequency}}$. For CER and UMASS, the number of rows is the number of days in the input image, and the wide frequency corresponds to the daily frequency which is 48 and 96 respectively. For DRED, the number of rows is the number of hours in the input image, and the wide frequency corresponds to the minute frequency which is 60. For example, to replicate the default configuration of the CER image using in [21], the resulting image shape would be $\mathbb{R}^{12 \times 48}$.

2) Data normalisation: This paper follows the necessary convention in normalising the values of input images. The input images are subject to a min-max normalisation to constrain input values between the range of $[0, 1]$. The normalisation is shown below:

$$x_i = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)} \tag{14}$$

C. Comparative Techniques

The primary benchmark for CRAE will be SCSAE as both models utilise a similar backbone as described in Section III-B. Other comparative techniques are described with their basic settings in Table III. The other comparative techniques were realised using sklearn [57], pywt [58] and scipy [59] packages.

D. Downstream Classification Task Design

The intention of the downstream task is to test the suitability of latent representations from CRAE, SCSAE and AE for direct application on a task thereby bypassing the decoder. Xiaoyao Huang, et al. [32] performed a similar procedure for their AE where the latent representation was used to predict the small-medium-enterprise status of a customer.

We design a downstream task which is more practical and has known real-world applications. Using DRED's dataset and auxiliary appliance-level data, the downstream task is to classify whether a microwave had been used in each hourly block. Such a task is akin to a lighter version of non-intrusive load management.

DRED's dataset provides auxiliary data with main meter readings and microwave-level outlet readings. We reshape the microwave-level outlet readings and classify each hourly block with more than 10 seconds of greater than 1000-watt readings as a Positive label, and then classify all remaining hourly blocks with Negative labels.

Finally, the encoded representations from all remaining non-encoder training DRED records are used. These encoded representations are split using 85% as downstream task training records and 15% as downstream task testing records. We train a Decision Tree algorithm and validate using the test records. The Decision Tree algorithm was chosen due to its low computational requirements and universal application.

TABLE IV
DEFAULT CONFIGURATION FOR SECTION V-A

Dataset	Wide Frequencies	Input Rows
CER	48	12
UMASS	96	12
DRED	60	60

V. EXPERIMENTAL RESULTS AND DISCUSSION

Three default configurations are used for the primary compression performance analysis in Section V-A and are detailed in Table IV. Further experimentation with input rows is detailed in Section V-B with a fixed ER of 12.

A. Compression Performance Analysis

The reconstruction errors at different ER levels are shown in Fig.6. Several key observations are pronounced: firstly, SCSAE for the CER dataset does not replicate results achieved by Shouxiang Wang, et al. [21]; secondly, CRAE has a lower reconstruction error compared against SCSAE across all ER for all datasets; thirdly, DWT (both one and two dimensional)

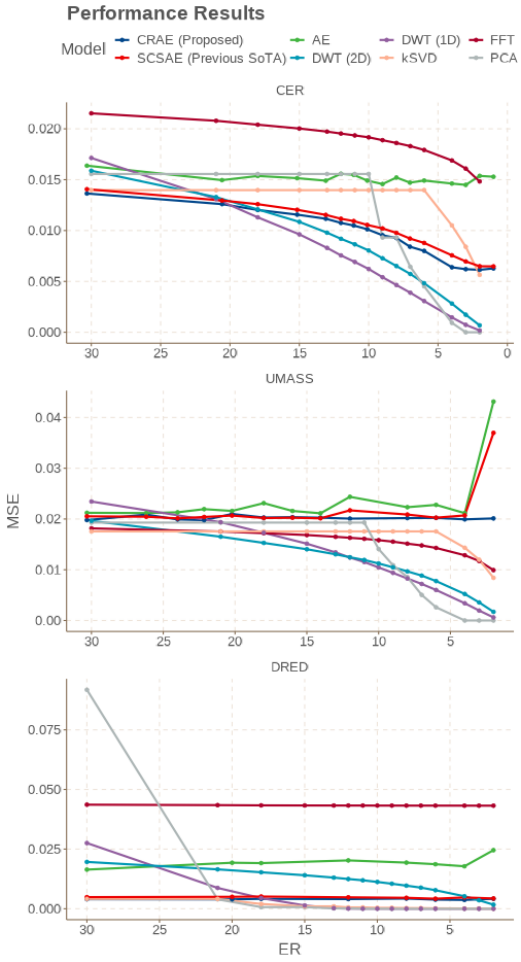

Fig. 6. Performance results across ER, using configurations detailed at IV

outperforms both SCSAE for both CER and UMASS; fourthly, CRAE and SCSAE do not appear to benefit from a higher ER for UMASS and DRED.

SCSAE for the CER dataset does not replicate results in [21]. The authors attained results which suggested that SCSAE was the state-of-the-art when compared to other methods, of which all are also applied in this paper. The key implementation differences between this paper and the SCSAE paper are primarily CER pre-processing differences and test set methodology. To begin, the authors only utilised $20,000$ complete daily blocks of shape $\mathbb{R}^{48}$ whereas in equivalent terms this paper utilises 108000 complete records of shape $\mathbb{R}^{48}$. In addition, CER contains both small and medium-sized enterprises (SME) and residential records. The authors, as part of their spectral clustering section, mention use of only SME records which may extend to their compression performance results. It is well documented that using inputs of similar distributions will allow for easier relationships to learn in deep learning, and in fact, this is leveraged by Jihoon Lee, et al. [33] who split datasets into high and low frequency to train two EED compression models, both with superior performance. This paper does not filter out any results by SME or residential status. Furthermore, the authors do not make any comment as to the training and testing split of the dataset. It is implied

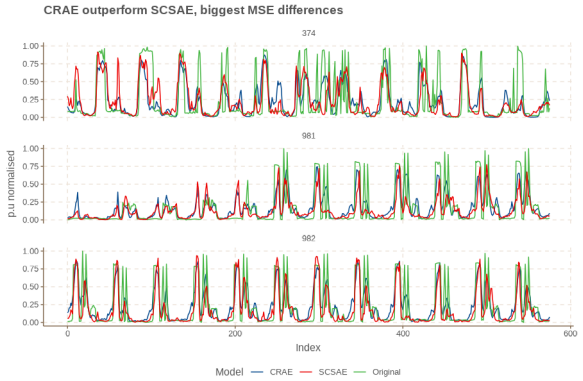

Fig. 7. Top 3 examples where CRAE outperforms SCSAE.

that the MSE reconstruction results achieved are those of the entire dataset, rather than a testing set. While this may be appropriate for application at the MDMS level where the entire dataset can be learned and encoded, the authors also iterate that the SCSAE was developed to be lightweight for smart meter application. A testing set must be used for a smart meter level application as it is unreasonable to expect that the smart meter will retrain the SCSAE on new data given limited computational resources. Thus, it is possible that this paper's results are attributable to consideration of the global and practical requirements of a smart meter compression algorithm. The implications from this are that AE-based networks are not competitive at many ERs.

Moreover, CRAE achieves a lower reconstruction error than SCSAE across all ER for all datasets. The performance increases are modest but do yield support to the theoretical benefits laid out in Section II and Section III. To investigate further, six reconstructed power curves from the CER testing set are presented. The three reconstructed power curves shown at Fig.7 are those which represent the largest performance increase of CRAE over SCSAE; and the three reconstructed curves shown at Fig.8. are those which show the largest performance degradation of CRAE from SCSAE. Upon comparison of the largest differences, it is evident that CRAE outperforms SCSAE when the power curves are more abrupt and volatile, whereas SCSAE outperforms CRAE when there are more stable recurring patterns. This behaviour is likely attributable to the inclusion of the LSTM layer, which results in the model being able to learn differing patterns between the days. The SCSAE model contrarily is able to learn local recurring patterns better and thus yields better results on the stable usage behaviours. This is line with expectations set out for this model. In summary, CRAE is more expressive than SCSAE. As it is likely that the more turbulent patterns appear than the stable patterns, CRAE will achieve a lower reconstruction error than SCSAE across ER for all datasets.

Additionally, DWT outperforms AE-based models for CER and UMASS. DWT was not a comparative algorithm at [21]. S. Ryu, et al. [34] also implemented a CAE and did compare its performance to DWT with the 'Haar' mother wavelet and level of decomposition was not disclosed. They found that CAE outperformed DWT. However, these implementation

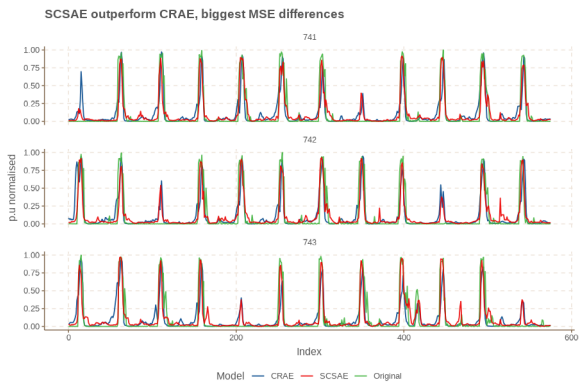

Fig. 8. Top 3 examples where SCSAE outperforms CRAE.

details for DWT are not optimal. B. A. Bhuiyan, et al. [19] studied the effect of wavelet and decomposition level on power compression and concluded that the 'db2' mother wavelet and decomposition level 3 was optimal for their dataset. Identically, this paper sought [19] for guidance and adopted the same hyperparameters and has found that DWT compression techniques, both one and two dimensional, outperform AE-based models. In seeking an explanation, we draw parallels to neural image compression and existing compression methods. Johannes Ball, et al. [15] studied the use of CAE with GDN activation functions that was optimised end-to-end including the quantizer which was fed into arithmetic coding. Even for an end-to-end network, the authors discovered that their results were not a significantly improvement on the commonplace JPEG2000 algorithm an algorithm which applies DWT to tile components of the image [61]. It was not until the introduction of the scale hyperprior in which neural compression significantly outperformed extant compression techniques [50]. This paper, and previous AE-based EED compression papers, are not end-to-end as neither quantization nor lossless compression is considered and thus lags image neural compression. This does leave a capacious pathway for future AE-based EED compression research. As such, the underperformance of SCSAE and CRAE against DWT is not spurious.

Lastly, neither SCSAE nor CRAE's reconstruction error appears to decrease with lower ER. As noted in Section III-B, minimal adjustments have been made to CRAE's layout to ensure that it remains comparable against SCSAE. As a result, all models use the same number of filters, filter sizes and LSTM hidden sizes despite the increase of input size. The change in the number of encoder parameters, which is discussed in Section V-C, is attributable to the first LSTM layer in the encoder as it accepts a larger dimension but still outputs the same hidden dimension size. As such, at higher frequencies, the logical deduction is that the latent dimension is responsible for a proportionally smaller explanation for the reconstructed output. As a primitive analogy, a latent dimension of 48 for the CER dataset would be 0.136% of the SCSAE encoder's parameters but would only represent 0.005% of SCSAE encoder's parameters for DRED. This analogy does not consider this paper's adjustments for input sizes, but still serves to illustrate the exponential increase in

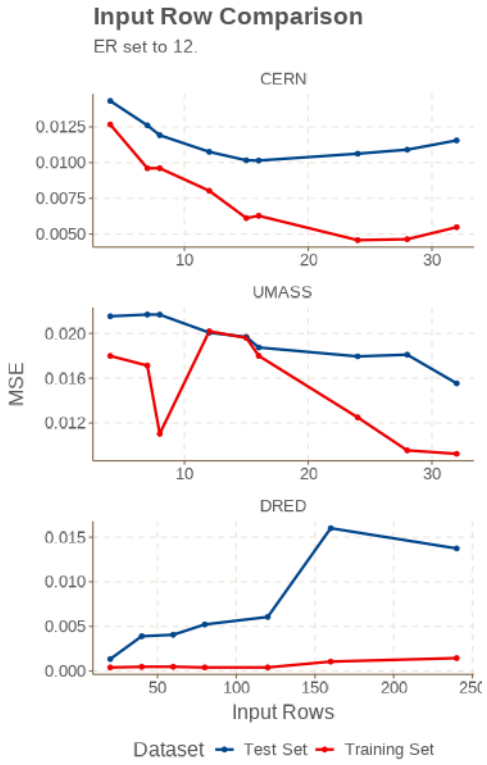

Fig. 9. Input row comparison, ER set to 12.

parameters relative to latent dimension size. As such, further research needs to be performed on expanded convolutional filter sizes and LSTM hidden sizes to increase model expressiveness such that the latent dimension has a noticeable impact on reconstruction errors. Nevertheless, CRAE and SCSAE's results for the DRED dataset are promising as these models outperform comparatives for ER greater than 22.

B. Input Row Analysis

To explore the impact of input rows and latent dimension size on the CRAE model, experiments have been performed and summarised at Fig.9. For these experiments, ER is set to 12. Thus, as input sizes are increased, latent dimension sizes correspondingly increased to maintain the ER. Two profound observations are evident: first, all three datasets behave remarkably differently in response to the input row changes; second, despite larger latent dimension sizes at higher input row sizes, this does not necessitate a decrease in MSE.

CERN, UMASS and DRED respond different to changes in input rows. CERN's test set displays parabolic behaviour with a local optimum at 14, while UMASS displays a monotonically decreasing test set and conversely DRED displays an increasing test set. This suggests that there a optimal input size which is different depending on the characteristics and the joint distribution of the dataset. Specifically, we interpret this to mean that the LSTM unit in the encoder is capable and actively learning relationships between the different input rows. For example, the local minimum for CERN at 14 implies that long-term fortnightly complexities do exist. Alternatively, UMASS' minimum at 32 inputs row implies that energy usage

TABLE V
ENCODER COMPUTATIONAL EFFICIENCY COMPARISON.

	Hyperparameters				Encoder Parameters			Memory Usage[a] (Kb)			CPU Runtime[b] (ms)		
Dataset	Frequency	Input Rows	Latent dimension d	ER	CRAE	SCSAE	Ratio	CRAE	SCSAE	Ratio	CRAE	SCSAE	Ratio
CER	48	12	48	12	47568	35152	1.35	124.64	52.31	2.38	9.71 ± 0.76	4.80 ± 1.46	2.03
UMASS	96	12	96	12	79872	118144	0.68	109.12	124.64	0.88	9.46 ± 0.44	2.27 ± 1.24	4.16
DRED	60	60	300	12	885324	948556	0.93	295.11	326.95	0.90	36.58 ± 15.64	3.02 ± 3.70	15.64

[a] Refers to peak memory usage; Peak memory usage is the same across all trials.
[b] CPU runtime for a batch size of 1 to emulate smart meter behaviour; Number of trials for each dataset and model combination: 50.

patterns at recurrent of up to at least 32 days and that the LSTM layer is capable of learning these patterns.

Conjointly, the prior beliefs of this paper's author is that an increase in input rows and correspondingly latent dimensions and model parameters would serve to monotonically decrease reconstruction error. This is evidentially untrue. Thus, it is implied that CRAE's architecture can be improved not by an increase in parameters, but by an increase in appropriate hyperparameter selection. As such, there is future research to be completed in proposing a methodology which can automatically learn the appropriate hyperparameter size. Alternatively, it is conceivable that the use of attention, either through transformers encoders [62] or directly as an extension of the LSTM layer [63], will eliminate the need to pre-emptively test and select for the optimal input rows, as the attention mechanisms will learn the appropriate input row level relationships themselves.

C. Computational Efficiency

The computational efficiency metrics comparing CRAE and SCSAE are shown at Table V. The experiments were performed on an Intel Xeon Gold 6150 CPU, as it is not expected that the smart meters will have GPU capabilities.

The results confirm that CRAE is markedly slower than SCSAE in terms of runtime for CER and DRED, but at higher frequencies CRAE utilises less peak memory than SCSAE. It must be noted that while differences are marked, they are not materially different. That is to say that while the runtime is proportionally larger, it is in absolute terms minute. To illustrate, SCSAE's average compression time for the hourly block of DRED data is 3.02ms and CRAE's corresponding average runtime is 36.58ms. Presumably, the hourly block of data will be encoded once agan hour. Therefore, the CRAE encoder will run for 36.58ms per hour which would appear to be satisfactory to an energy operator. Assuming that the CPU is 100 times slower, an extrapolation of the runtime will still only take 36.58s per hour which is still acceptable.

However, the results do indicate that the relative scalability of CRAE is poorer than SCSAE. It can be observed that the ratio of CRAE to SCSAE CPU Runtime is increasing as the input size and latent dimension size increases. These results are unsurprising given that the LSTM layer has encode serially and is not capable of threading or other parallelisation techniques. In fact, transformers, mentioned in Section V-B, were designed to forgo the serial calculation and transfigure it into quadratic time complexity. Therefore, if the input size of the energy image increases to an extent that is not operable on a smart meter, consideration should be given to Transformer encoders (though counterproductively also their restrictive memory requirements). Nevertheless, the computational requirements are currently satisfactory based on these three case studies.

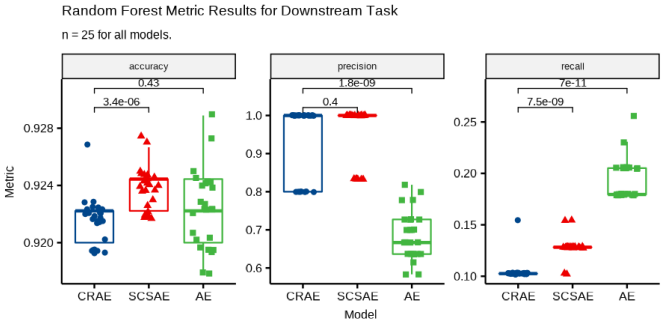

Fig. 10. Boxplot of Classification Task Results, $n = 25$

D. Downstream Classification Task Results

The results of the downstream task are shown in Fig.10. The random forests are trained 25 times for each model to ensure that the random intialisations and subsets are thoroughly accounted for, and to yield statistical power to the test. The p-values shown in the braces are based on the Kruskal-Wallis one-way analysis of variance, or H-test. The H-test is a non-parametric test of mean ranks, which requires fewer assumptions than ANOVA.

The results indicate that the accuracy and recall from the CRAE encoding classifiers are distributionally different from the SCSAE encoding classifiers in a statistically significant manner ($p = 3.4 \times 10^{-06}$ and $p = 7.5 \times 10^{-09}$ respectively). As such, if direct encodings are required or beneficial, SCSAE encodings yield superior performance.

The results are somewhat surprising given that the CRAE encodings can be reconstructed into the original energy image with lower reconstruction error than SCSAE encodings, as discussed in Section V-A. We do not have any hypothesis as to this phenomenon though do suggest that a future research pathway would be to investigate whether an end-to-end CRAE

model outperforms an end-to-end SCSAE model. If so, then the results here may suggest that CRAE's encodings are 'overfit' for the downstream task, but are tuned purely for reconstruction.

VI. Conclusion

This paper has presented a novel hybrid deep learning EED compression technique. Through an unsupervised learning approach, the CRAE model is able to exploit the intrinsic local and long-term dependencies of energy data. The proposed method is validated with three different datasets. Compared with the existing SCSAE model, the reconstruction error has reduced modestly while satisfying computation requirements, making it more appropriate for smart meter level application. In particular, the CRAE achieves a lower reconstruction error on more 'volatile' usage patterns than SCSAE. Notably, we observe that DWT achieves lower reconstruction error across a substantial number of ER settings raising the performance benchmark.

The direct application of CRAE's latent representations on a downstream classification task was explored. We identify that the existing SCSAE's representations outperform CRAE.

In future work, the CRAE model will be explored in an end-to-end fashion with quantization and arithmetic encoding. Furthermore, the encoder and decoder will be augmented with attention mechanisms to learn the most appropriate long-term dependencies.

Acknowledgement

I'd like to thank Dr. Hao Wang for his thought-provoking advice and guidance he has provided me through this thesis journey.

References

[1] 2022. [Online]. Available: https://www.eia.gov/tools/faqs/faq.php?id=108\&t=3#:~:text=How%20many%20smart%20meters%20are,installations%20were%20residential%20customer%20installations

[2] 2015. [Online]. Available: https://www.audit.vic.gov.au/report/realising-benefits-smart-meters?section=

[3] S. Aggarwal and N. Kumar, *Chapter Twenty-Three - Smart grid*. Elsevier, 2021, vol. 121, pp. 455–481. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0065245820300784

[4] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 112–121, 2014.

[5] M. F. Elahe, M. Jin, and P. Zeng, "Review of load data analytics using deep learning in smart grids: Open load datasets, methodologies, and application challenges," *International journal of energy research*, vol. 45, no. 10, pp. 14 274–14 305, 2021.

[6] A. Unterweger and D. Engel, "Lossless compression of high-frequency voltage and current data in smart grids," in *2016 IEEE International Conference on Big Data (Big Data)*, Conference Proceedings, pp. 3131–3139.

[7] A. Abuadbba, I. Khalil, and X. Yu, "Gaussian approximation-based lossless compression of smart meter readings," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5047–5056, 2018.

[8] Y. Wang, Q. Chen, C. Kang, Q. Xia, and M. Luo, "Sparse and redundant representation-based smart meter data compression and pattern extraction," *IEEE Transactions on Power Systems*, vol. 32, no. 3, pp. 2142–2151, 2017.

[9] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," *IEEE transactions on smart grid*, vol. 10, no. 3, pp. 3125–3148, 2019.

[10] P. Zhang, H. Cheng, B. Zou, P. Dai, and C. Ye, "Load data mining based on deep learning method," in *ACM International Conference Proceeding Series*. ACM, Conference Proceedings, pp. 1–5.

[11] [Online]. Available: https://webstore.iec.ch/publication/6398

[12] L. Wen, K. Zhou, S. Yang, and L. Li, "Compression of smart meter big data: A survey," *Renewable and Sustainable Energy Reviews*, vol. 91, pp. 59–69, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032118301849

[13] Y. Yang, S. Mandt, and L. Theis, "An introduction to neural data compression," *arXiv preprint arXiv:2202.06533*, 2022.

[14] K. Brandenburg, "Mp3 and aac explained," in *Audio Engineering Society Conference: 17th International Conference: High-Quality Audio Coding*. Audio Engineering Society, Conference Proceedings.

[15] J. Ball, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," *arXiv preprint arXiv:1611.01704*, 2016.

[16] J. C. S. de Souza, T. M. L. Assis, and B. C. Pal, "Data compression in smart distribution systems via singular value decomposition," *IEEE transactions on smart grid*, vol. 8, no. 1, pp. 275–284, 2017.

[17] R. Mehra, N. Bhatt, F. Kazi, and N. Singh, "Analysis of pca based compression and denoising of smart grid data under normal and fault conditions," in *2013 IEEE International Conference on Electronics, Computing and Communication Technologies*. IEEE, Conference Proceedings, pp. 1–6.

[18] R. Li, F. Li, and N. D. Smith, "Load characterization and low-order approximation for smart metering data in the spectral domain," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 976–984, 2017.

[19] B. A. Bhuiyan, M. W. Absar, and A. Roy, "Performance comparison of various wavelets in compression of pmu generated data in smart grid," in *2017 3rd International Conference on Electrical Information and Communication Technology (EICT)*, Conference Proceedings, pp. 1–6.

[20] P. Liu, P. Zheng, and Z. Chen, "Deep learning with stacked denoising auto-encoder for short-term electric load forecasting," *Energies (Basel)*, vol. 12, no. 12, p. 2445, 2019.

[21] S. Wang, H. Chen, L. Wu, and J. Wang, "A novel smart meter data compression method via stacked convolutional sparse auto-encoder," *International journal of electrical power & energy systems*, vol. 118, p. 105761, 2020.

[22] H. Wang, G. Henri, C. W. Tan, and R. Rajagopal, "Activity detection and modeling using smart meter data: Concept and case studies," in *2020 IEEE Power & Energy Society General Meeting (PESGM)*, Conference Proceedings, pp. 1–5.

[23] M. Nabil, M. Ismail, M. Mahmoud, M. Shahin, K. Qaraqe, and E. Serpedin, "Deep recurrent electricity theft detection in ami networks with random tuning of hyper-parameters," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, Conference Proceedings, pp. 740–745.

[24] M. N. Fekri, H. Patel, K. Grolinger, and V. Sharma, "Deep learning for load forecasting with smart meter data: Online adaptive recurrent neural network," *Applied Energy*, vol. 282, p. 116177, 2021.

[25] F. nal, A. Almalaq, and S. Ekici, "A novel load forecasting approach based on smart meter data using advance preprocessing and hybrid deep learning," *Applied Sciences*, vol. 11, no. 6, p. 2742, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/6/2742

[26] J. Cho, Z. Hu, and M. Sartipi, "Non-intrusive a/c load disaggregation using deep learning," in *2018 IEEE/PES Transmission and Distribution Conference and Exposition (T&D)*. IEEE, Conference Proceedings, pp. 1–5.

[27] R. U. Madhure, R. Raman, and S. K. Singh, "Cnn-lstm based electricity theft detector in advanced metering infrastructure," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, Conference Proceedings, pp. 1–6.

[28] Z. A. Khan, T. Hussain, A. Ullah, S. Rho, M. Lee, and S. W. Baik, "Towards efficient electricity forecasting in residential and commercial buildings: A novel hybrid cnn with a lstm-ae based framework," *Sensors*, vol. 20, no. 5, 2020. [Online]. Available: ⟨GotoISI⟩://WOS:000525271500161

[29] T.-Y. Kim and S.-B. Cho, "Predicting residential energy consumption using cnn-lstm neural networks," *Energy*, vol. 182, pp. 72–81, 2019.

[30] G. Xun and G. Yan, "Non-intrusive load decomposition method based on cnn-lstm model," in *International Conference on Electronic Information Engineering and Computer Technology (EIECT 2021)*, vol. 12087. SPIE, Conference Proceedings, pp. 477–483.

[31] X. Zhou, J. Feng, and Y. Li, "Non-intrusive load decomposition based on cnnlstm hybrid deep learning model," *Energy Reports*, vol. 7, pp. 5762–5771, 2021.

[32] X. Huang, T. Hu, C. Ye, G. Xu, X. Wang, and L. Chen, "Electric load data compression and classification based on deep stacked auto-encoders," *Energies*, vol. 12, no. 4, 2019.

[33] J. Lee, S. Yoon, and E. Hwang, "Frequency selective auto-encoder for smart meter data compression," *Sensors*, vol. 21, no. 4, 2021.

[34] S. Ryu, H. Choi, H. Lee, and H. Kim, "Convolutional autoencoder based feature extraction and clustering for customer load analysis," *IEEE Transactions on Power Systems*, vol. 35, no. 2, pp. 1048–1060, 2020.

[35] Y. Yuan, Q. Zhang, K. Dehghanpour, F. Bu, and Z. Wang, "Smart meter data compression and reconstruction using deep convolutional autoencoders," in *2020 52nd North American Power Symposium (NAPS)*. IEEE, Conference Proceedings, pp. 1–5.

[36] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *2010 IEEE Computer Society Conference on computer vision and pattern recognition*. IEEE, Conference Proceedings, pp. 2528–2535.

[37] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[38] G. Toderici, D. Vincent, N. Johnston, S. Jin Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, Conference Proceedings, pp. 5306–5314.

[39] E. Dasan and I. Panneerselvam, "A novel dimensionality reduction approach for ecg signal via convolutional denoising autoencoder with lstm," *Biomedical Signal Processing and Control*, vol. 63, p. 102225, 2021.

[40] R. Mutegeki and D. S. Han, "A cnn-lstm approach to human activity recognition," in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Conference Proceedings, pp. 362–366.

[41] C. Wang, F. Jiang, and H. Yang, "A hybrid framework for text modeling with convolutional rnn," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, Conference Proceedings, pp. 2061–2069.

[42] X. Shi, T. Wang, L. Wang, H. Liu, and N. Yan, "Hybrid convolutional recurrent neural networks outperform cnn and rnn in task-state eeg detection for parkinson's disease," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, Conference Proceedings, pp. 939–944.

[43] J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, "Dimensional sentiment analysis using a regional cnn-lstm model," in *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*, Conference Proceedings, pp. 225–230.

[44] J. Ball, V. Laparra, and E. P. Simoncelli, "Density modeling of images using a generalized normalization transformation," *arXiv preprint arXiv:1511.06281*, 2015.

[45] P. T. Yamak, L. Yujian, and P. K. Gadosey, "A comparison between arima, lstm, and gru for time series forecasting," in *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, Conference Proceedings, pp. 49–55.

[46] M. R. Raza, W. Hussain, and J. M. Merig, "Cloud sentiment accuracy comparison using rnn, lstm and gru," in *2021 Innovations in intelligent systems and applications conference (ASYU)*. IEEE, Conference Proceedings, pp. 1–5.

[47] S. Yang, X. Yu, and Y. Zhou, "Lstm and gru neural network performance comparison study: Taking yelp review dataset as an example," in *2020 International workshop on electronic communication and artificial intelligence (IWECAI)*. IEEE, Conference Proceedings, pp. 98–101.

[48] A. Shewalkar, "Performance evaluation of deep neural networks applied to speech recognition: Rnn, lstm and gru," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 9, no. 4, pp. 235–245, 2019.

[49] A. N. Shewalkar, "Comparison of rnn, lstm and gru on speech recognition data," 2018.

[50] J. Ball, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *arXiv preprint arXiv:1802.01436*, 2018.

[51] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[52] L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE winter conference on applications of computer vision (WACV)*. IEEE, Conference Proceedings, pp. 464–472.

[53] A. Unterweger, D. Engel, and M. Ringwelski, "The effect of data granularity on load data compression," in *Energy Informatics*, S. Gottwalt, L. Knig, and H. Schmeck, Eds. Springer International Publishing, Conference Proceedings, pp. 69–80.

[54] 2012. [Online]. Available: https://www.ucd.ie/issda/data/commissionforenergyregulationcer/

[55] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, and J. Albrecht, "Smart*: An open data set and tools for enabling research in sustainable homes," *SustKDD, August*, vol. 111, no. 112, p. 108, 2012.

[56] p. 4554, 2015. [Online]. Available: https://doi.org/10.1145/2821650.2821659

[57] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Dubourg, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[58] G. Lee, R. Gommers, F. Waselewski, K. Wohlfahrt, and A. O'Leary, "Pywavelets: A python package for wavelet analysis," *Journal of Open Source Software*, vol. 4, no. 36, p. 1237, 2019.

[59] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, and J. Bright, "Scipy 1.0: fundamental algorithms for scientific computing in python," *Nature methods*, vol. 17, no. 3, pp. 261–272, 2020.

[60] Y. Wang, Q. Chen, C. Kang, Q. Xia, Y. Tan, Z. Zeng, and M. Luo, "Residential smart meter data compression and pattern extraction via non-negative k-svd," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*. IEEE, Conference Proceedings, pp. 1–5.

[61] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The jpeg2000 still image coding system: an overview," *IEEE transactions on consumer electronics*, vol. 46, no. 4, pp. 1103–1127, 2000.

[62] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, . Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[63] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based lstm for aspect-level sentiment classification," in *Proceedings of the 2016 conference on empirical methods in natural language processing*, Conference Proceedings, pp. 606–615.