Robust Data Driven Analysis for Electricity Theft Attack-Resilient Power Grid

Inam Ullah Khan[1], Nadeem Javeid[2], C. James Taylor[1] and Xiandong Ma[1,*]
[1]*Engineering Department, Lancaster University, Bailrigg, Lancaster LA1 4YW, UK*
[2]*Department of Computer Science, COMSATS University Islamabad, Pakistan*
*Correspondence: xiandong.ma@lancaster.ac.uk

***Abstract*—The role of electricity theft detection (ETD) is critical to maintain cost-efficiency in smart grids. However, existing ETD methods cannot efficiently handle the sheer volume of data now available, being limited by issues such as missing values, high variance and non-linearity. An integrated infrastructure is also required for synchronizing diverse procedures in electricity theft classification. To help address such problems, a novel ETD framework is proposed that combines three distinct modules. The first module handles missing values, outliers, and unstandardised electricity consumption data. The second module employs a newly proposed hybrid class balancing approach to deal with highly imbalanced datasets. The third module utilises an improved artificial neural network (iANN) based classification engine, to predict electricity theft cases accurately and efficiently. We propose three distinctive mechanisms, including hyper-parameters tuning, regularization and skip connections, to improve the performance of standard ANN to handle more complex classification tasks using smart meter (SM) data. Furthermore, various structures of iANN are investigated to improve the generalization and function fitting capabilities of the final classification. Numerical results from real-world energy usage datasets confirm that the proposed ETD model has superior performance compared to existing machine learning and deep learning methods, and can effectively be applied to industrial applications.**

***Index Terms*—Smart grid, Smart meter data, Classification, Electricity theft detection.**

I. INTRODUCTION

Energy crises are real, extensive and seem to be long-lasting. This is neither inevitable nor desirable. During the transfer of energy, power system networks encounter two types of losses: technical losses (TL) and non-technical losses (NTL) [1]. TL are inherent and cannot be averted because of their occurrence in transformers, cables and long-distance transmission lines during the transfer of energy. NTL has long plagued the utilities and has two dominant components, namely electricity theft and non-payment of utility bills.

Electricity theft with its many facets usually has an enormous cost to utilities compared to non-payment because of energy wastage and power quality problems. It has always been a problem for power utilities and no electric power utility is immune to power theft. Today, it is estimated that electricity theft costs the power industry as much as $96 billion/year globally. In developing countries, this proportion is much higher, with an estimated cost of $60 billion/year [2]. This huge loss drives up prices for end-users, increases the need for costly government subsidies, and cripples utility companies around the globe.

One of the main aims of the smart grid is to lower power system losses to equate the electricity demand-supply gap. With the recognition of the internet of things (IoT) technologies and data-driven approaches (based on single-level data collection), power utilities have enough tools to combat electricity theft and fraud. The electricity consumption changes frequently and a large amount of installed IoT devices monitor the multi-source real-time data, such as climatic factors (wind, solar, temperature), transmission and the consumers' electricity usage record. For example, during the uncertain times of COVID-19, when people could be spending more time indoors, the quantity of historical data is big and difficult to analyse [3], [4].

II. RELATED WORK AND CONTRIBUTIONS

Machine learning (ML), deep learning (DL) and time-series models are the main approaches for electricity theft detection (ETD) in smart grid. Based on smart meter data, normal and abnormal power consumption patterns and footprints can be identified with irregular, longer and higher electricity usage patterns than regular and normal consumption. The ML algorithms are gradually trained based on supervised learning to determine the relationship between input features (consumption) and corresponding labels (field inspection results). The work described by [4]–[7] concerns supervised ML algorithms to characterize the class label of normal and anomalous power consumption patterns. Since these algorithms utilize already fabricated data, the computational cost is moderate with no requirement for new hardware devices and prior knowledge about network topology. However, there are several shortcomings in existing classification-based schemes, such as the high false-positive rate (FPR), time-consuming engagement of experts, and low adaption to new types of electricity fraud [2].

Given the importance of boosting and DL algorithms, a limited but growing body of literatures [8]–[11] utilized the publicly available SGCC (State Grid Corporation of China) dataset and successfully applied for NTL detection in smart grid. Hussain *et al.* [8] used a feature engineered based category boosting (CatBoost) algorithm in conjunction with the SMOTETomek sampling algorithm for ETD. The proposed model achieved an area under the curve (AUC) score of 92%. However, it is very challenging for boosting algorithms to attain a higher accuracy due to the presence of the various outliers, noise and data sparsity since each estimator in boosting

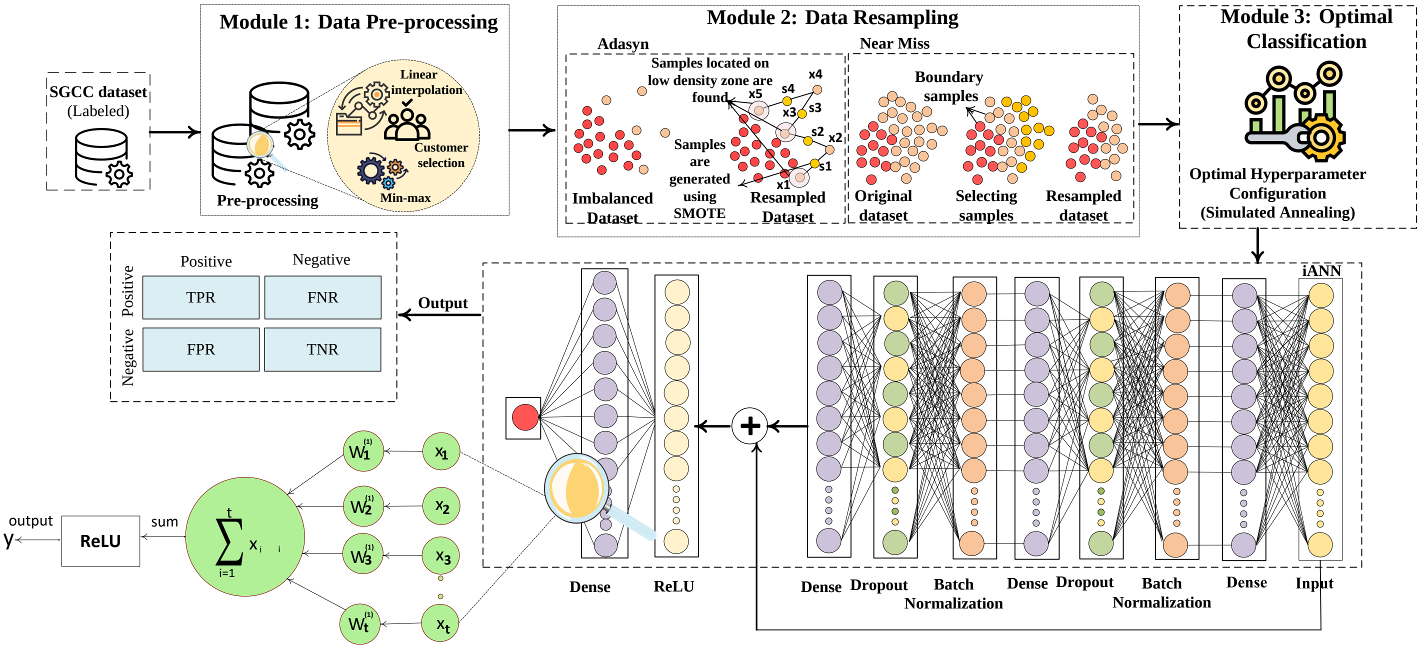


Fig. 1: Proposed electricity theft detection framework

algorithms is obliged to fix the error of the predecessors. Study in [9] exploited a CNN based long short term memory (LSTM) model for ETD. In this proposed hybrid model, CNN is used to automate the feature extraction process, whereas LSTM is used to solve a classification problem. The authors also utilized the synthetic minority oversampling technique (SMOTE) to avoid class imbalance problem. However, SMOTE algorithm generates synthetic data instances for minority class samples to obtain an equal distribution of both majority and minority samples. It causes low generalization and overfitting problems, resulting in inaccurate prediction model results for unseen/test data. In [10], the authors proposed a DL methodology based on multilayer perceptron (MLP) and a convolutional neural network (CNN) to capture electricity theft from raw electricity consumption (EC) data. However, a major drawback of using CNN and MLP networks is their difficulty in handling large time series data. Due to this, the input is limited to a fixed size window and the prediction model cannot capture a descent in the EC data if it occurred before the analysis period. More recent work in [11] utilized a deep siamese network (DSN) to discriminate between honest and dishonest consumers in EC data. The proposed model achieved good prediction results but at the cost of two shortcomings, as compared to the other well performing DL methods [12]. First, DSNs are relatively slow to train due to quadratic pairs learning. Secondly, the output of DSN does not involve probabilities due to involvement of the pairwise learning, hence making it not generalizable and sensitive to some variations in the input [13].

Time-series data analysis methods are widely used in ETD. For example, autoregressive integrated moving average (ARIMA) have shown good performance in stable electricity markets. In this regard, Singh *et al.* [14] proposed a relative entropy concept that captures variations in probability distribution obtained from multiple consumers. Similarly, Jokar *et al.* [15] made use of energy consumption patterns as a base recognition system to model the predictability of normal and abnormal consumption patterns with advanced metering infrastructure (AMI). Although statistical methods help capture the partial non-stationariness in recorded data and could be critical for ETD, the presence of various outliers and building the model on raw data may make the classification accuracy unstable.

Macro-level (micro-grid) and micro-level (smart meter) energy consumption profiles are fundamental to the application of the classifier. It is essential to enrich the attributes of normal energy consumers and differentiate the outliers to relate to the energy theft phenomenon. In a binary classification problem, various aggregating methods are also used for ETD. In a recent work, Jindal *et al.* [16] proposed energy consumption data aggregation for multiple households in local communities. For households, the authors employed a decision tree (DT) algorithm to predict the energy consumption value and, subsequently, a support vector machine (SVM) classifier was trained on multiple features to locate customers with anomalous consumption behaviour. On a similar task, Pulz *et al.* [17] used census data to extract social indicators to find the interdependence between losses and socio-economic indices for ETD under various scenarios. Such aggregated data-driven approaches are useful; however, problems like non-stationary high-volume data measurements need to be addressed to compose useful clusters.

Mostly, the aforementioned literature focuses on classifier design or feature engineering algorithms, where conventional classifiers, e.g., SVM and DT algorithms are popular [3], [19].

However, SVM usually has a high computational cost and it is a challenge to find optimal values of hyper-parameters to achieve higher classification results. DT, on the other hand, possesses over-fitting problems that mean its performance is high during training (seen data) but not in prediction (unseen data). Besides, ML and DL methods rarely take big data into account and the experiments are conducted considering load or price data, which is not sufficient. Thus, theft detection precision could still be improved considering big data from grid sources.

A. *Contributions*

In this work, we examine binary classification issues for ETD in smart grids. Our objective is to predict the honest and dishonest consumers accurately using big data from the smart grid. To achieve this challenging task, we propose an improved artificial neural network (ANN) for the underpinning framework that performs energy theft tasks efficiently with normal and anomalous consumption patterns. Compared to the shallow ML methods, we preferred to choose ANN for the classification task because it has stronger non-linear computational and complex function abilities. Also, it is more suitable for classification tasks due to many potential advantages to learning essential laws and key features from mass data. An ANN is formed when neural structures are constituted in the form of layers. The computational power of a neural network is attained by connecting hundreds of single-unit artificial neurons with their respective weights. The artificial neuron, a processing element, has weighted inputs and an output associated with a transfer function. Although ANN is a promising approach, the subsequent challenges need to be addressed to predict electricity theft with higher accuracy:

- *Challenge 1 (Highly imbalanced theft data)*: In real datasets, data samples are not represented in equal proportion. It is the scenario when the fraudulent instances far outweigh non-fraud instances. Standard methods to tackle imbalanced class problems are random over-sampling and under-sampling. However, due to certain known drawbacks, the classifier is biased towards majority class samples (honest consumers) and shows inaccurate performance for minority class samples (fraudulent in our case). In binary class problems, the accurate classification of minority class samples is more important to handle.
- *Challenge 2 (High computational complexity)*: The DL methods are slow to train. According to e.g. [20], the neural networks' performance is constrained by processing uncertain pieces of information. Also, these methods have high computational costs due to the operation of forward and backward propagations through the hidden layers. In the ETD process, irrelevant and repetitious features make the classifier training procedure challenging and prevent it from being a good fit model, which ultimately lowers the final prediction outcome of the classifier.
- *Challenge 3 (Problem of limited generalization and over-fitting)*: One major difficulty with training deep architectures is exploding and vanishing gradients. As back propagation computes gradients using the chain rule, gradients can exponentially grow or vanish, preventing weights from updating and thus stalling training. Another issue faced by neural networks is the internal covariate shift (ICS) which occurs when the distribution of network activation changes because of variations in network parameters during training. As ANNs have a large number of layers, this shift in input distribution can be problematic in achieving fast convergence. Also, ANNs have the most common problems of over fitting, limited generalization and limited control over convergence and stability.

To address the above mentioned challenges and to assist electrical utilities to identify energy fraud, we develop a novel ETD framework, called sequential preprocessing, resampling and classification (SPRC), as presented in Fig. 1. The main components of SPRC are sequential preprocessing based on interpolation, outliers handling and standardization (IOS), hybrid data resampler (HDR) and final classification with improved ANN (iANN). Precisely, an interpolation method fills missing values in the dataset to attain data uniformity. Afterwards, operations like outliers handling and normalization are performed to make data consistent and set data values between 0-1. Like any other real-world data, electricity theft data also contain primarily samples of honest (91%>) consumers and very few data samples are of fraudulent (9%<) consumers. Thus, we develop an HDR based an adaptive synthetic (ADASYN) oversampling and near-miss under-sampling (NMU) technique to obtain balance distribution for classifier training. Once the data are in well-organized shape, the processed data are sent to the iANN for final classification. In the proposed framework, we also propose different iANN structures (sequential, parallel and other combinations) to improve the generalization and better function capabilities of the classifier. In contrast to relying on the output of a single structure, it is expected that numerous mixes of iANN structures would give higher prediction performance. We also proposed an integrated preprocessing approach in one of our recent conference papers and showed some initial results [21]. The current work is built on the same concept but uses a new procedure for resampling and classifier design and, more importantly, we investigate different configurations of the iANN and new methods to improve the ANN performance. In particular, to achieve higher accuracy and computational efficiency, this paper makes the following improvements:

- First, an IOS-based data preparation module employs data imputation, outliers handling and standardization algorithms to ensure data accuracy and critical insights. This helps reduce human error during inspections, such as typos or overlooked items missed by the human eye. Secondly, an HDR combines the advantages of over-sampling and under-sampling techniques to avoid the severely skewed class distribution problem for real-world datasets. Finally, a multi-mode classification engine, based on iANN, is designed to complete the prediction task. The

ANN's performance is improved by adopting different procedures such as hyper-parameters tuning, regularization methods and skip connections (HRS). The HRS-ANN has significantly better performance than many ML and DL methods proposed in this field. Moreover, among the different structures of the multi-mode classification engine (iANN), the most effective structure is chosen for the final classification.
- For performance assessment of the proposed methods, extensive experiments have been conducted on real-world data traces from the electric grid's workload. The simulation results reveal that the presented method achieves better classification results than existing approaches proposed in this field.

The remainder of this paper is organised as follows. Section III presents the data preparation and class balancing modules. In section IV, the ANN and its improvement methods are presented. Section V verifies the proposed framework with experimental results. Finally, section VI concludes this work.

III. SYSTEM FRAMEWORK

The primary issue in ETD methods is to maximize classification accuracy. However, various factors affect the electricity consumption pattern and make the classifier training process difficult and complex. To enhance proposed framework accuracy, we propose a sequential IOS, a newly developed HDR for class balancing and an HRS-ANN-based improved classification method. As shown in Fig.1, the SPRC procedure starts with ordering and standardizing the raw data. The standardization methods are essential for the implementation of the whole framework under consideration. Secondly, the standardised data are fed into the class balancer to handle class imbalance issues. Finally, the prepared data are sent to develop the ANN. Since ANN performance depends on several hyper-parameters, we employ the simulated annealing (SA) algorithm to tune these parameters. Furthermore, we use regularization methods such as batch normalization, early stopping and weight decay for addressing the dual challenges of generalization and computational efficiency.

It is well-established that neural network performance degrades when more hidden layers are added to the network [22]. However, the addition of hidden layers is essential when handling large datasets in ETD. The addition of extra layers offers better opportunities to learn hierarchical re-composition of complex features. To avoid the degradation problem, we propose the use of a skip connection-based ANN to improve classifier accuracy. Finally, learned from [23], the most effective topology of multi-mode iANN is utilized for theft prediction. A detailed explanation of these modules is given in the following sections.

A. *Data Preparations*

Data preparation is often the first and most essential step when analysing electricity consumption data for a specific problem. This section describes the process of data preparation for which we apply a sequential IOS method on the collected data. This includes data imputation, outlier handling and data standardization (data centring and scaling). We assume a matrix

$$\text{X} = \begin{bmatrix} x_{11} & x_{12} & ... & x_{1n} \\ x_{21} & x_{22} & ... & x_{2n} \\ \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ x_{m1} & x_{m2} & ... & x_{mn} \end{bmatrix} = \begin{bmatrix} \overrightarrow{t_1} \\ \overrightarrow{t_2} \\ . \\ . \\ . \\ \overrightarrow{t_m} \end{bmatrix}, \tag{1}$$

where

$$\overrightarrow{t_k} = [x_{k1}, x_{k2}, ...x_{kn}]\, k \in [1, m]. \tag{2}$$

represents the electricity consumption pattern. The rows and columns depict the time stamps and the feature index of recorded data, respectively. The index, i.e., x_{mn} is the $n-th$ component of the $m-th$ power usage values that require classification.

1) ***Recovering missing data****:* Due to various reasons, the recorded data often have missing values. Some of the associated reasons are failure of hardware, storage issues, unscheduled maintenance, unreliable transmission of measurement data and data corruption. In the present work, the unknown (missed values) are recovered using an interpolation method [10] based on,

$$f(x_i) = \begin{cases} \left(\frac{x_{i-1}+x_{i+1}}{2}\right), & if\ x_i \in NaN, x_{i\pm1} \notin NaN, \\ x_i, & otherwise, \end{cases} \tag{3}$$

where x_i is a missed (null) recorded value represented as NaN.

2) ***Handling outliers****:* The presence of outliers increases data variability and distorts real results. The "three-sigma rule of thumb" introduced in [10] is used to deal with outliers as follows,

$$f(x_i) = \begin{cases} X, & if\ x_i > X, \\ x_i, & otherwise, \end{cases} \tag{4}$$

where X is a vector that consists of multiple entries of x_i and can be computed as $Avg(X) + 2\sigma(X)$. $Avg(X)$ and $\sigma(X)$ represent the average value and standard deviation of X.

3) ***Data standardization****:* Often, attributes in historic data comprises of different scales. We apply the MIN-MAX scaling method to rescale all the values to the range 0-to-1 as follows,

$$x_{new} = \frac{x_i - min(x)}{max(x) - min(x)}. \tag{5}$$

B. *Hybrid Data Resampler*

One of the critical problems which require particular attention in real-world electricity theft datasets is the unequal data distribution or domination of the majority class (honest samples) over the minority class (fraud samples). Due to the imbalanced data distribution of target class, the classifier gets skewed towards majority class samples and better learn key characteristics and features belonging to the majority class [24]. As a result, minority samples are most often left unattended, and hence the classifier shows inaccurate prediction results towards fraud cases (minority class samples).

To address the above-mentioned issue, a strategic combination of both over-sampling and under-sampling techniques are proposed to reduce the misclassification cost of minority samples. The proposed novel method is named HDR, and it is used for the first time to solve the unequal class distribution problem in this framework design.

In HDR, ADASYN [25] and NMU [26] are employed sequentially. First, ADASYN synthetically generates alternatives (not duplicates) for each observation of the minority class. Let m_j and m_i be the observations of majority and minority class samples respectively, such that $m_i \leq m_j$ and $m_i + m_j = m$. The degree of imbalanced ratio is calculated using $d = \frac{m_i}{m_j}$. The cumulative number of synthetic samples required for the minority class is determined as $G = (m_j - m_i) \times \beta$. The variable β represents the desired balanced level of minority and majority samples after applying ADASYN. An ideal situation arises when $\beta = 1$, meaning that the minority and majority samples are equal. For each observation of the minority class, $x_i \in m_i$, the k nearest numbers are obtained based on Euclidean distance to calculate the ratio $r_i = \frac{majority\ samples}{k}$. After normalizing the density distribution $\hat{r_i} = \frac{r_i}{\sum r_i}$, the synthetic samples to generate per neighbourhood are calculated using $g_i = \hat{r_i} \times G$. Finally, synthetic data alternatives S_i are generated using the following equation,

$$S_i = x_i + \lambda(x_k - x_i) \tag{6}$$

where variable λ represents a random number $\lambda \in$ [0, 1] and $(x_k - x_i)$ is the difference vector in n dimensional space. Unlike ADASYN, the NMU is based on the nearest neighbour algorithm with multiple variants to remove unnecessary majority class observations from class boundaries. First, the number of majority and minority class observations are counted. Secondly, the average distance of majority class observations to each minority class observation $d(m_i, m_j)$ is calculated based on their Euclidean distances. Finally, each minority class observation picks three closest k nearest majority class observations in the majority class. The resampled dataset has only those majority class observations which have the least distance with minority observations in the feature space and discards the others. This procedure repeats until the algorithm achieves a uniform distribution for both classes.

Note that the efficiency of the iANN classifier in terms of ADASYN, NMU and HDR (ADASYN+NMU) is evaluated in Section V-C.

IV. CLASSIFIER ADJUSTMENT

After the two stages of data preparations and resampling, the data are in a standardised form to train the classifier. This section provides a detailed description of our proposal to accomplish the final classification task. Since the ANN is robust and efficient enough for supervised learning tasks, we choose ANN as the classifier.

A. *Problem Formulation*

In this paper, the classification problem is modelled to compute binary cross entropy loss between actual and predicted class using the following equation,

$$L = -\frac{1}{N}\left[\sum_{i=1}^{N} y_i \times log(h_\theta(x_i)) + (1 - y_i)\ log(1 - h_\theta(x_i))\right] \tag{7}$$

where N and y_i denote training samples and true class value for the input-output pair (x_i, y_i). The non-linear hypothesis $h_\theta(x)$ of the neural network is calculated below,

$$h_\theta(x) = f(w^T x + b), \tag{8}$$

where w and b represent weights and biases to train the model, and the activation function is denoted by $f(.) : \mathbb{R} \rightarrow \mathbb{R}$. Compared to the conventional logistic sigmoid function and hyperbolic tangent, we prefer rectified linear unit (ReLU) $f(z) = max\{0, z\}$ to increase the ANN learning rate. For a given sample, the output value (activation) of unit i in layer k is defined as follows,

$$\begin{aligned} a_i^k = f(z_i^k) = f(w_{i_1}^{k-1} a_1^{k-1} + w_{i_2}^{k-1} a_2^{k-1} \\ + ... + w_{ip_{k-1}}^{k-1} a_{p_{k-1}}^{k-1} + b_i^{k-1}) \end{aligned} \tag{9}$$

where z_i^k denotes the weighted sum of all activations a_i^k, p_k denotes the number of neurons in layer k. Similarly, input layer K_1 and output layer K_{n_k} units activation are computed as,

$$a_i^1 = x_i, \tag{10}$$

$$\begin{aligned} h_\theta(x) = a_i^{n_k} = f(w_{i_1}^{n_k-1} a_1^{n_k-1} + w_{i_2}^{n_k-1} a_2^{n_k-1} \\ + ... + w_{ip_{n_k-1}}^{n_k-1} a_{p_{n_k-1}}^{n_k-1} + b_i^{n_k-1}). \end{aligned} \tag{11}$$

The activations of each unit in the input, output and hidden layers are computed using forward propagation. The objective is to minimize L by adjusting the trainable parameters w and b using a stochastic gradient descent (SGD) algorithm. For this purpose, first small random values (near zero) of w_{ij}^k and b_i^k are initialized and forward propagation computes the activation of each unit from the first hidden layer towards the final layer. In every iteration of the SGD algorithm, each parameter is updated in order to minimise the loss as follows,

$$w_{ij}^k = w_{ij}^k - \alpha\ \frac{\partial\ L(w,b)}{\partial\ w_{ij}^k} \tag{12}$$

$$b_i^k = b_i^k - \alpha\ \frac{\partial\ L(w,b)}{\partial\ b_i^k}, \tag{13}$$

where α represents the learning rate. We apply back-propagation to compute the partial derivatives and update each weight in the network, thereby minimizing the error for each output neuron and the network as a whole. The back-propagation algorithm is based on four fundamental steps to compute the error (δ^k) and the gradient of the cost function [27].

1) First, the forward propagation computes the activation of each unit in layer K_2 up to the layer K_{n_k}.

TABLE I: ANN hyper-parameters using simulated annealing

Hyper-parameter	Range of values	Optimal value
Activation	Tanh, Relu, Sigmoid	Relu
Batch_size	15, 30, 45, 60, 75, 90	60
Solver	Sgd, Adam, Nadam	Sgd
Alpha	0.0001, 0.003, 0.05, 0.07	0.05
Learning_rate	Constant, Adaptive	Adaptive

2) Calculate the residual (error) for each unit i in layer n_k,

$$\delta_i^{n_k} = \frac{\partial}{\partial\ z_i^{n_k}} |y_i - h_\theta(x_i)| = -(y_i - a_i^{n_k}) \acute{f}(z_i^{n_k}). \quad (14)$$

3) Calculate the residual in each unit i in layer k, k = n_k -1, n_k – 2, . . . , 2,

$$\delta_i^k = (\sum_j^{p_{k+1}} w_{ji}^k \delta_j^{k+1}) \acute{f}(z_i^k). \quad (15)$$

4) Calculate the partial derivatives with respect to w and b,

$$a_j^k \delta_i^{k+1} = \frac{\partial\ L(w,b)}{\partial\ w_{ij}^k}, \qquad \delta_i^{k+1} = \frac{\partial\ L(w,b)}{\partial\ b_i^k}. \quad (16)$$

5) Finally, weight updating to minimise the error,

$$\Delta w_{ij}^k = -\frac{\alpha\ L(w,b)}{\partial w_{ij}^k}. \quad (17)$$

With the process of back forward and iterative steps of SGD, the neural network is trained to decrease the cost function in Eq. 7.

B. Optimal Classification

As discussed before, the main objective of this framework design is to minimize the loss function given in Eq. 7. However, there exists a strong relationship among the loss function and ANN hyper-parameters, which are the number of hidden layers, activation function, batch size and learning rate. It is hard to obtain optimal values of hyper-parameters to improve accuracy and efficiency. The conventional methods adopted for the adjustment of ANN's hyper-parameters are the SGD algorithm or cross-validation [18]. However, the adoption of these two methods may lead to higher computational costs and convergence problems. DL models are computationally expensive. According to [28], DL models are approaching computational limits. The researchers discovered that DL models advancement has been "strongly reliant" on increased computational power. They asserted that continued progress would require "dramatically" more efficient computational DL methods, either through modifications to existing methods or new as-yet-undiscovered procedures. In SPRC, therefore, HRS methods are applied for optimal classification. These methods are described below.

1. Simulated annealing-based ANN: For practical and computationally hard optimization problems, the SA algorithm is preferred over exact algorithms such as gradient descent [29]. The main inspiration behind the algorithm operation is annealing mechanism in metallurgy that fist applies the heating process followed by a gradual cooling procedure of substance to obtain defect-free crystals [29]. The task is performed in three steps: initialization, transition mechanism for diverse states, and finally, the cooling schedule composed of an objective function with multiple variables. The elements in the SA algorithm are represented by a vector that contains hyperparameter values for optimization. This procedure is repeated unless the optimal values of all hyper-parameters given in Tables I and II are obtained. It is pertinent to mention that important hyper-parameters as well as their initial values for ML models (used later for comparison in Section V) and iANN are borrowed from Ref. [2]. The four main steps of the SA algorithm are as follows:

i. Start with random initialization of population.

ii. At each iteration, evaluate more suitable solution considering the fitness (objective) function.

iii. Selection of new solution based on a probability-based decision whether to discard or retain current solution.

iv. Progressive decrease in temperature from a maximum to the minimum (zero) values. An inadequate solution receives a zero moving probability, while a positive moving probability is assigned to the adequate solution.

For parameter tuning, a hyper-parameters application programming interface (API) is used to configure hyper-parameters automatically [31]. The optimization toolkit is highly adaptable to perform model optimization for different preprocessing and classification modules. Contrary to the traditional tedious search methods, it searches the best combination of hyper-parameters in an automated manner and can therefore outperform human professionals and experts in algorithms design.

2. The role of regularization: Regularizations are the process of modifying a learning algorithm to prevent over-fitting. Regularizers help limit the learning process to a subset of the hypothesis space with manageable complexity. With the adoption of modern regularization techniques such as batch normalization, early stopping and weight decay to penalize large weights, the effective Rademacher complexity of the possible solutions is dramatically reduced [32].

2A). Batch normalization accelerates the learning process of deep ANN and reduces ICS problem and generalization error. It stabilizes the initial random weights and configuration of the learning algorithm to achieve a stable distribution of activation throughout training [33]. ICS of activation i at time t is defined as the difference,

$$||G_{t,i} - G'_{t,i}||_2 \quad (18)$$

$$G_{t,i} = \Delta_{w_i^{(t)}} L(w_1^{(t)}, ..., w_k^{(t)}) \quad (19)$$

$$G'_{t,i} = \Delta_{w_i^{(t)}} L(w_1^{(t+1)}, ..., w_{i-1}^{(t+1)}, w_i^{(t)}, w_{i+1}^{(t)}, ..., w_k^{(t)}) \quad (20)$$

where L is loss, w_1^t , . . . , w^t are the parameters of each n_k layers, $G_{t,i}$ corresponds to the gradient of the layer parameters, and $G'_{t,i}$ is the same gradient after all the previous layers have been updated with their new values.

2B). An early stopping technique is incorporated into the training process, which not only prevents over-fitting but helps train a model with fewer epochs [34]. It is a form of regulariza-

TABLE II: Hyper-parameters of the benchmark models

Classifier	Main hyper-parameters	Values range	Obtained values

LR	Penalty strength (C), Penalty (R).	C = 0.001, 0.01, 10, 100.
		R = L_1 norm, L_2 norm.
			C = 0.01,
			R = L_2 norm.

RF	Number of sample leaves (SL),
	Decision trees (DT),
	Sample splits (SS),
	Criterion.
		SL = 3, 5, 8, 11, 15.
		DT = 5, 10, 20, 30, 40.
		SS = 5, 6, 7, 8, 9, 10.
		Criterion = gini, entropy.
			SL = 5, DT = 20,
			SS = 7,
			Criterion = gini.

SVM	Kernel function (K),
	Loss function (σ),
	Penalty (C).
		K= linear, rbf, sigmoid, poly.
		σ = 0.001, 0.01, 0.1, 1, 10.
		C = 0.01, 0.1, 1, 10, 100.
			K= rbf,
			σ = 0.01,
			C = 1.

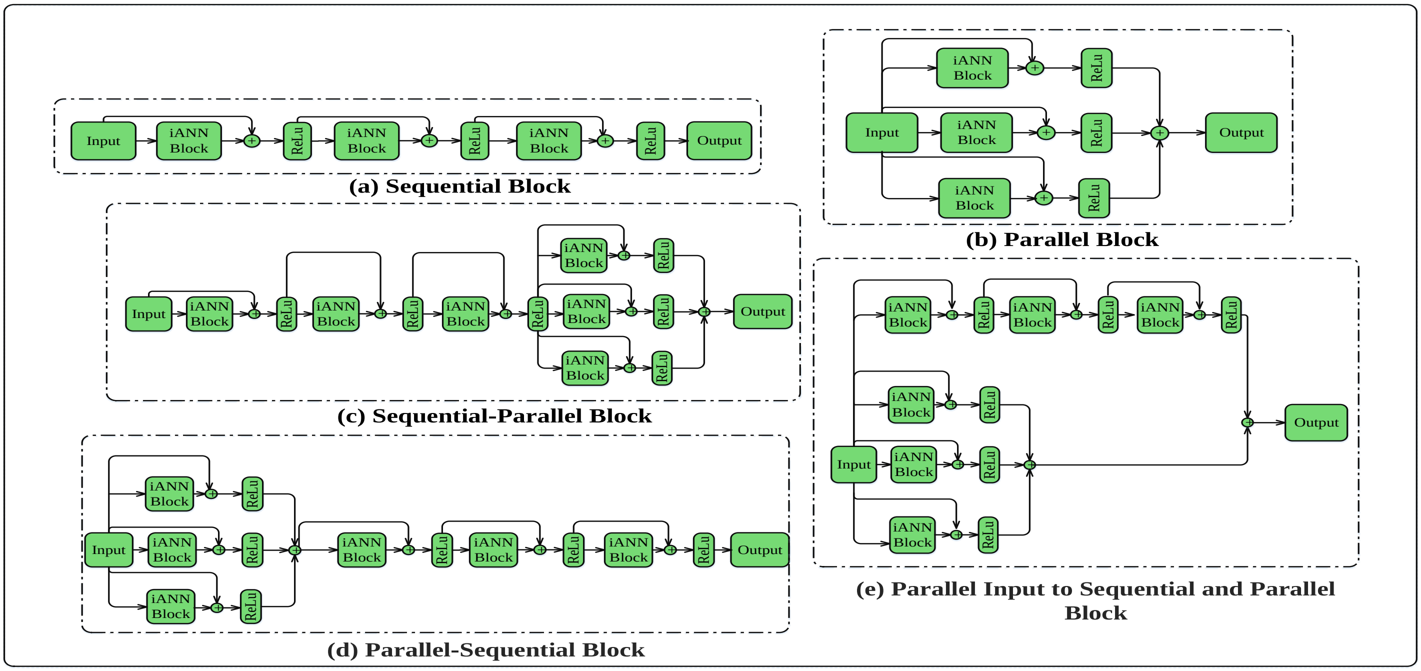

Input
iANN Block
ReLu
Output
(a) Sequential Block
(b) Parallel Block
(c) Sequential-Parallel Block
(d) Parallel-Sequential Block
(e) Parallel Input to Sequential and Parallel Block

Fig. 2: Structure of the proposed prediction engine; (a): Sequential block, (b): Parallel block, (c): Sequential-Parallel block, (d) Parallel-Sequential block, (e): Parallel input to Sequential and Parallel block

tion that allows an arbitrarily large number of training epochs and terminates the training process when model performance stops improving.

2C). Weight decay is a well-established regularization technique to keep neural network weights small and avoid an exploding gradient [35]. The general formula for updating the weights as follows,

$$w_i^{t+1} = w_i^t - \eta \frac{\partial L}{\partial w_i} - \mu \Delta w_i^{t-1}, \tag{21}$$

where η and μ represent learning rate and momentum terms in the ANN, respectively. The simple addition of a regularization term to prevent over-fitting and to constrain the magnitude of the weights is as follows,

$$w_i^t = w_i^t - \eta \frac{\partial L}{\partial w_i} - \mu \Delta w_i^{t-1} - \Upsilon w_i^t. \tag{22}$$

where Υ is a weight decay parameter to control the relative importance of regularization. When Υ = 0, the weight decay property can be easily disabled to obtain typical behaviour.

Cross-validation (CV) is a standard well-performing method for generalized model performance evaluation. The SGCC dataset has a relatively high imbalance distribution of the target class values and using traditional k-fold CV may lead to inconsistent test results [36]. We use stratified k-fold CV (SCV), an advanced version of k-fold CV, to obtain an equal distribution of both classes. As a result, a small difference between the testing performance of the model is obtained. Guided by feedback from the SCV used in our experiments, we received very reliable estimates when using 6-fold SCV for iANN.

3. Role of skip connections: The original intuition "the deeper the better" is not always useful to learn complex features and representations. A research team at Microsoft [22] investigated the relationship between depth and network performance and established that the percentage error for a 56-layer network is higher than a 20-layer network on both training and testing data. This problem of training very deep networks has been addressed to a greater extent with recently developed residual neural networks (ResNets) [37]. ResNets feature residual or skip connections to distribute learning behaviour across layers, display minimum decay in gradients and make the training of individual residual blocks easier. In ResNets, a direct connection skips some layers (this may vary in different models) in between and connects directly to the output. This connection is called ‘skip connection’ and is the core of residual blocks. The overall representation of the residual block becomes,

$$X_{l+1} = \Psi(F_l(x_l) + x_l) \tag{23}$$

where F_l represents the residual function and $\Psi(x)$ is the ReLU activation $max(0, x)$.

C. *Multi-Block Classification Engine*

Enlightened by the findings of [23], we develop various iANN based classification engines and extensive experiments have been conducted to achieve higher convergence accuracy and time management. All variables in the classification engine are optimized either using the regularization method described in Section IV-B or with rigorous trial and error to increase the training mechanism and classification engine precision. Moreover, we implement various models of the suggested classification engine, based on iANN, with numerous mixes such as the sequential/cascade framework, sequential-parallel, parallel-sequential and combined parallel construction, as illustrated in Fig. 2, in order to choose the best-combined approach.

Fig. 2a shows the sequence of the serial iANN blocks. First, the standardised data are provided to the first iANN block as an input and the predicted results of this particular block are given to the next block. The main goal of the model is to fit the error through performance enhancement. Similarly, Fig. 2b presents the parallel mode of iANN combinations. The sequence of these blocks is very important to form different connections. As seen from the figure, the same input at the same time is considered by all blocks. Also, the same output will be evaluated by this structure and aggregated as the process result.

Fig. 2c-2e presents the extended building blocks of the structures mentioned with different topologies. The exogenous values such as load, price and related parameters in the time series data are provided to the classification engine as an input in the form of a matrix. The performance of the extended structures can be enhanced by assigning higher weights to the best presentation and by no or low weightings to the weak networks.

With the integration of IOS, HDR and HRS-ANN, the electricity theft prediction approach can classify fraudulent activity accurately. The next section explains experiments and analyses based on illustrative real-world theft data.

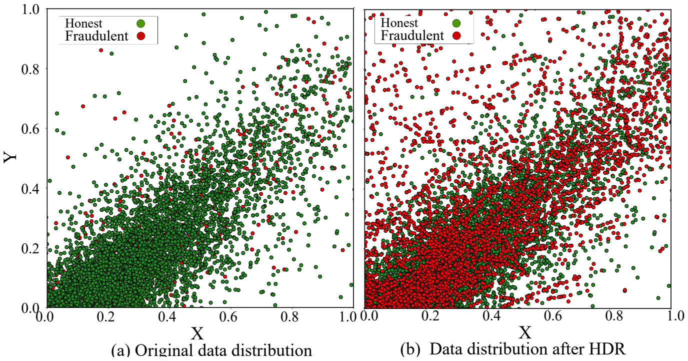
V. EXPERIMENTAL RESULTS

A. *Case Study Setup and Data Availability*

For performance evaluation, five different case studies are implemented in Google Co-laboratory in accordance with the system framework devised in Section III. The load profile data of 42372 consumers is obtained from SGCC for 1035 days i.e., from 2014 to 2016. Here, 38757 consumers are recognized as honest and the 3615 consumers as dishonest, as shown in Table III. The models are trained and tested on actual SM data. The SGCC dataset is the only publicly available labeled dataset with at least one on-field inspection [38] . The data have been divided into a training and a test dataset to generalize model capabilities beyond the training/seen dataset. The division is performed in a stratified manner so that there is the same percentage (%) of NTL samples in the training and test datasets. The dataset used for training purposes consists of 80% of the labeled data, while the test dataset consists of 20%.

TABLE III: Metadata information

Description	Value
Time window	01-01-2014 to 31-10-2016
Resolution of data	Daily data
Number of consumers	42372
Number of days (features)	1035
Fair consumers	38757
Unfair consumers	3615
Consumers type	Residential
Source type (RES, conventional)	Utility



(a) Original data distribution (b) Data distribution after HDR

Fig. 3: Data representations before and after handling imbalanced class

B. *Performance Metrics*

There are four expected outcome values in a confusion matrix (CM) from a binary classifier i.e., true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). Based on CM results, Accuracy, Precision, Recall and F1-score performance metrics are computed below,

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}, \tag{24}$$

$$Precision = \frac{TP}{TP+FP}, \tag{25}$$

$$Recall = \frac{TP}{TP+FN}, \tag{26}$$

$$F_1\ Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \tag{27}$$

The AUC score is often chosen as an evaluation metric to evaluate classification accuracy. It provides more reliable assessment when the class distribution is highly imbalanced. The mathematical formula for AUC calculation is as follows [10],

$$AUC = \frac{\Sigma_{i \in PC} Rank_i - \frac{M(1+M)}{2}}{M \times N} \tag{28}$$

where M and N are the number of positive and negative instances in the positive class (PC), and $Rank_i$ is the rank value of a sample i in an ascending order. The value of AUC highlights that the probability of choosing a positive number is relatively higher then choosing a randomly negative number. The curve is a graphical representation the true positive rate (TPR) and false positive rate (FPR) plotted on the y-and x-axes, respectively.

$$True\ positive\ rate = \frac{TP}{TP+FN} \tag{29}$$

$$False\ positive\ rate = \frac{FP}{FP+TN} \tag{30}$$

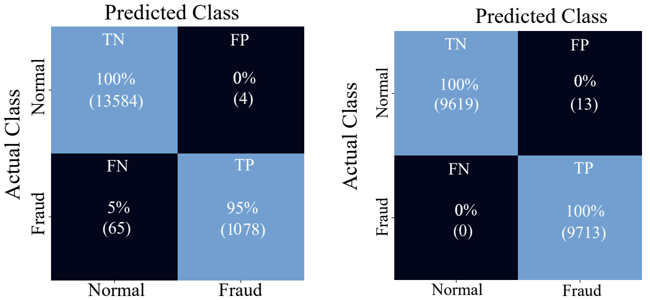
The TPR is the fraction of positive classes labelled correctly while the FPR represents the fraction of negative class samples that are misclassified. Notably, the higher the AUC, the better the classifier's performance. When the AUC tends straight up to the maximum value and then turn towards the x-axis, it indicates that both classes are distinguished perfectly by the classifier [19]. By contrast, when AUC = 0.5 and the curve point tends towards the diagonal line, this yields that the classifier has no power to discriminate between both classes.

C. Performance Results

*1) **Performance of data balancing module:*** In this section, we empirically study the effects of no sampling, oversampling, under-sampling and HDR on the final classification. Figs. 3a and 3b show the presence of minority and majority classes of data samples before and after handling the imbalanced class problem. The majority class samples (green circles) are in much greater number, as shown in Fig. 3a, and a biased classification is expected because the classifier is trained more on negative samples. Without handling the highly imbalanced data distribution problem, Fig. 4a displays a severe performance loss when classifying fraudulent users, whereas the values of TN, FN, FP and TP are 100%, 5%, 0% and 95%, respectively. The honest customers, TN, are identified 100% correctly; however, the value of FN is much higher, which means the classifier incorrectly indicates dishonest consumers as honest.

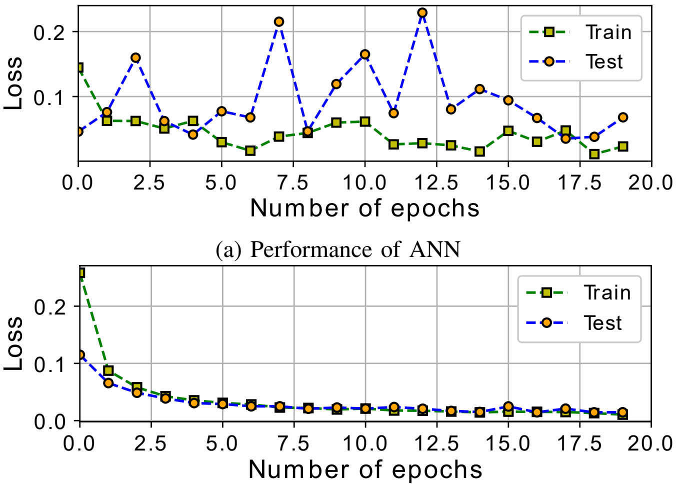
In ETD, the FN value needs to be reduced because these consumers are the real culprits who indulged in illegal usage of electricity. To resolve this issue, we utilize HDR, which efficiently obtains a balanced distribution for minority and majority classes, as shown in Fig. 3b. The balanced data distribution improves model training as well as generalization capabilities. The improved numerical results are given in the form of the CM in Fig. 4b.

*2) **iANN performance comparison with ANN:*** We compare the performance of iANN with standard ANN and the results are shown in Figs. 5 and 6. Fig. 5 shows the loss (how bad or good prediction is) graph for ANN and iANN only for twenty epochs. The irregular upper plot for ANN in Fig. 5a reveals that the prediction results both for training and testing datasets have small granular feedback on performance. Also, the standard ANN has issues like overfitting and generalization errors, due to which it shows unstable performance for test/unseen data. The lower plot for iANN in Fig. 5b reveals that the training process converges well and the loss is smooth between the probability distributions. Parameter tuning has a big impact on model training as it correlates model convergence, model accuracy, infrastructure resource requirements (as a result of cost) and training time. In Fig. 6, the AUC score for iANN is 97.9% compared to the ANN, which has only 93.6%. The superior performance of iANN mainly comes from the integration of improvement techniques in DL areas. It jointly employs HRS first to optimize the hyper-parameters of the ANN, followed by regularization methods to resolve over-fitting problems and finally skip connection to distribute the learning behaviour across the layers. It is pertinent to mention here that the use of the SA algorithm increases computational time. However, we apply two newly developed methods: regularization and skip connections. Due to the combined effects of these diverse but interconnected procedures, both higher accuracy and reduction in computational complexity are achieved simultaneously.



(a) Imbalanced class (b) Balanced class

Fig. 4: Prediction results before and after resampling



(a) Performance of ANN

(b) Performance of iANN

Fig. 5: Learning curves

*3) **Performance comparison of different multi-block classification engines:*** This case study employs five different topologies of iANN, and the one best performing model is selected as the classification engine. Fig. 6 illustrates AUC curves for RF, LR, SVM, ANN and iANN with the proposed

TABLE IV: Comparison among different modes of classification engine

Classifier	Accuracy	Precision	Recall	F1-Score	AUC	Training Time
Sequential	0.994	0.996	0.966	0.981	0.966	3min 55s
Parallel	0.996	0.996	0.978	0.987	0.978	2min 36s
Par_Seq	0.997	0.996	0.987	0.991	0.987	4min 12s
Seq_Par	0.996	0.996	0.983	0.989	0.983	5min 59s
Par_Seq_Par	0.995	0.995	0.973	0.984	0.973	4min 42s

TABLE V: Robustness comparison among SPRC and other benchmark schemes

Classifer	Training Ratio 60%				Training Ratio 70%				Training Ratio 80%			
	Precision	Recall	F1-Score	AUC	Precision	Recall	F1-Score	AUC	Precision	Recall	F1-Score	AUC
LR	0.550	0.538	0.733	0.624	0.827	0.862	0.875	0.820	0.951	0.954	0.955	0.941
RF	0.573	0.577	0.654	0.641	0.748	0.748	0.733	0.701	0.774	0.771	0.767	0.720
SVM	0.637	0.654	0.664	0.690	0.688	0.689	0.689	0.684	0.773	0.688	0.747	0.719
ANN	0.748	0.744	0.816	0.781	0.793	0.795	0.855	0.878	0.856	0.865	0.947	0.936
CNN-LSTM [9]	0.664	0.615	0.661	0.666	0.629	0.662	0.636	0.670	0.670	0.690	0.676	0.730
WD-CNN [10]	0.640	0.691	0.651	0.689	0.624	0.720	0.770	0.718	0.661	0.760	0.685	0.756
DSN [11]	0.875	0.839	0.857	0.860	0.840	0.850	0.845	0.844	0.912	0.923	0.928	0.934
iANN	0.947	0.945	0.943	0.934	0.941	0.947	0.961	0.958	0.964	0.954	0.982	0.979
Par_Ser (Proposed)	0.950	0.950	0.973	0.938	0.979	0.978	0.979	0.968	0.996	0.987	0.991	0.987

TABLE VI: Benchmark frameworks

Benchmark	Description
SPRC (Proposed)	IOS + HDR + Par_Seq
E	IOS + HDR + iANN
D	IOS + HDR + ANN
C	IOS + ADASYN [25] + ANN
B	IOS + NMU [26] + ANN
A	Without IOS and Resampling

TABLE VII: Computational time vs accuracy

Classifier name	Time (seconds))	AUC score
LR	230	0.941
RF	130	0.720
SVM	241	0.719
ANN	198	0.936
iANN	232	0.979
Proposed (SPRC)	282	0.987

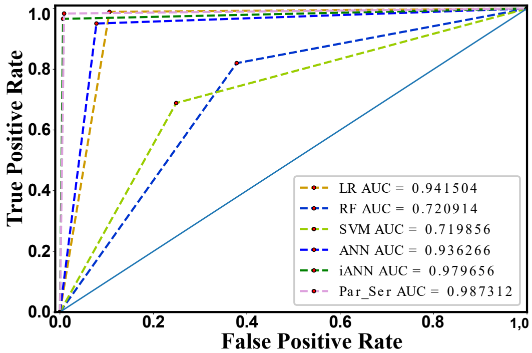


Fig. 6: AUC score for different structures of the multimode classification engine

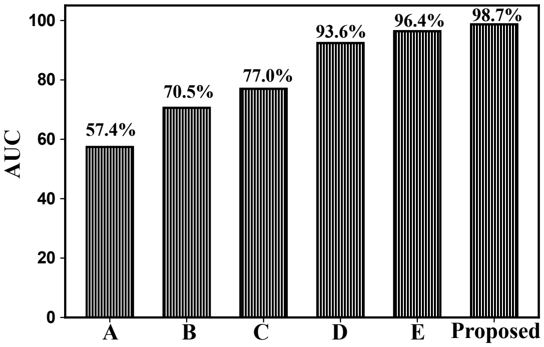


Fig. 7: Comparison of accuracy among SPRC and benchmark frameworks

topology. Tables I and II show the range of values searched as well as the optimal value found with the SA algorithm. It is seen in Table IV that the results obtained from combined topologies of iANN are comparable. The standalone sequential and parallel topologies, however, tend to obtain weak classification results because of the over-fitting problem (and other possible reasons associated with ANN as described in Section II-A). The results in Table IV show the superiority of the parallel_sequential (Par_Seq) and sequential_parallel (Seq_Par) structures. However, we select Par_Seq topology as the final classifier to guarantee reduced computational complexity, higher accuracy and robustness of the prediction results.

*4) **The SPRC robustness comparison with benchmark algorithms**:* Robustness is the ability of a network to perform well when it is subject to failures. The main aim of this case study is to examine whether SPRC guarantees network robustness under multiple scenarios. First, a random noise (Jitter) is added to each input pattern during network training. The addition of noise is attained via the Gaussian Noise layer in Keras (the software library used). The layer requires the standard deviation of the noise to be specified as a parameter. In this way, time-series patterns are recycled to explicitly learn robust features and the average accuracy of the algorithm is observed. Thus, deliberately introducing noise is one way to help hold our models accountable.

The second way is to observe the model's performance on different proportions of training data. The difference is subtle. A small dataset can cause the network to memorize all training examples. We seek to have them learn the characteristics of training data and not memorize them. In essence, DL model performance is severely affected by the size of input/training data. The aim is to confirm whether SPRC maintains its superiority when small (60%), medium (70%) and high sizes (80%) of training samples, compared to the size of all samples, are used as input to train the classifier. The experimental results in Table V illustrate that the SPRC achieves higher prediction accuracies for all sizes of the training dataset compared to the other algorithms under consideration. It is notable that the conventional schemes adopt an expanding trend when more data are available for training. For these data, the SPRC attains a maximum AUC score of 0.987 and surpasses the other well-known algorithms in terms of performance metrics. Furthermore, under a similar training/testing dataset ratio, the comparison results in Table V have shown that the proposed model can surpass the performance of other state-of-the-art methods such as CNN-LSTM [9], WD-CNN [10] and DSN [11] due to the reasons as discussed in Section II.

*5) **The SPRC performance on theft detection**:* This case study compares the theft detection performance of the SPRC approach with other benchmark approaches. The benchmarks considered for this investigation are given in Table VI. As displayed in Fig. 6, the SPRC has a higher AUC score for electricity theft prediction of this data set than all the benchmarks. The comparison among frameworks A, B, C, D, E and SPRC in Fig. 7 suggests that every module with the relevant description we proposed, can increase the accuracy of electricity theft prediction. The classifier learns the problem much faster if we can better expose the structure to the network for learning. The SPRC prepares quality data with IOS followed by HDR to curb the class imbalance problem. The hyper-parameter tuning, regularizations and skip connections we proposed improve the ANN performance, hence ensuring higher accuracy of electricity theft prediction. From Table VII, it is noticed that the results from LR and standard ANN are comparable; however, the RF and SVM models are unable to distinguish fair and fraud electricity consumptions patterns. This is because RF usually faces overfitting problems and SVM performance degrades when large datasets are used for training purposes. Also, when computational times of the proposed method and other benchmark algorithms are compared, the SPRC takes time to complete the classification task at a comparable level with LR, SVM and iANN. There exists a trade-off between higher accuracy and computational time. More accurate algorithms are generally more computationally expensive and vice versa.

VI. CONCLUSIONS

We have investigated how a highly imbalanced class distribution dataset can be arranged to train a classifier for the identification of normal and abnormal electricity consumption patterns. The presented approach integrates data preprocessing, resampling and multi-stage classification modules into a single model. The classification is comprised of a multi-block neural network that is optimized by an intelligent algorithm, regularization methods and skip connection to increase model training and classification abilities. Moreover, different multi-block prediction models were presented to choose the effective model. The proposed topologies have been applied over real-world data with a number of cases studied. We found that tuning the classifier's hyper-parameters with an intelligent algorithm results in smoother optimization and reduced computational complexity of the learning process. Similarly, regularization methods help to reduce the over-fitting and ICS problems associated with the standard ANN. We found that residual networks distribute learning across layers, each of which is responsible for learning better representations, while standard networks concentrate on learning in shallower layers and thus do not make effective use of deeper layers.

The above is supported by results for gradient norms, where non-decaying gradients are observed during training and testing in terms of robustness. These results show that varied training rates in SPRS do not change the representation as much as for the benchmark algorithms. In addition, we find that the parallel-sequential topology is more robust to varied learning rates.

In the next step, we will perform three further investigations to improve the performance of SPRC in terms of robustness and scalability. First, we will exploit knowledge from power grid sources, network distribution topology and geographic information to monitor energy consumption pattern abnormalities. Secondly, the average accuracy of the classifier in terms of robustness will be investigated adding random noise (Jitter) and synthetically generated theft attacks on selected data. Thirdly, the SPRC performance will be tested on unsupervised publicly available datasets. For this purpose, synthetic data will be generated to label the dataset and make it useful for supervised learning.

Acknowledgements The authors acknowledge funding support from Lancaster University, U.K., and COMSATS University Islamabad to support this project. The work was in part supported by the U.K. Engineering and Physical Sciences Research Council (EPSRC) grant EP/V051059/1.

REFERENCES

[1] K. Zheng, Q. Chen, Y. Wang, C. Kang, and Q. Xia, "A novel combined data-driven approach for electricity theft detection," IEEE Transactions on Industrial Informatics, vol. 15, no. 3, pp. 1809-1819, 2018.

[2] M.-M. Buzau, J. Tejedor-Aguilera, P. Cruz-Romero, and A. Gómez-Expósito, "Hybrid deep neural networks for detection of non-technical losses in electricity smart meters," IEEE Transactions on Power Systems, vol. 35, no. 2, pp. 1254-1263, 2019.

[3] K. Wang, C. Xu, Y. Zhang, S. Guo, and A. Y. Zomaya, "Robust big data analytics for electricity price forecasting in the smart grid," IEEE Transactions on Big Data, vol. 5, no. 1, pp. 34-45, 2017.

[4] N. F. Avila, G. Figueroa, and C.-C. Chu, "NTL detection in electric distribution systems using the maximal overlap discrete wavelet-packet transform and random undersampling boosting," IEEE Transactions on Power Systems, vol. 33, no. 6, pp. 7171-7180, 2018.

[5] J. I. Guerrero, I. Monedero, F. Biscarri, J. Biscarri, R. Millan, and C. Leon, "Non-technical losses reduction by improving the inspections

accuracy in a power utility," IEEE Transactions on Power Systems, vol. 33, no. 2, pp. 1209-1218, 2017.

[6] M. M. Buzau, J. Tejedor-Aguilera, P. Cruz-Romero, and A. Gómez-Expósito, "Detection of non-technical losses using smart meter data and supervised learning," IEEE Transactions on Smart Grid, vol. 10, no. 3, pp. 2661-2670, 2018.

[7] E. W. S. Angelos, O. R. Saavedra, O. A. C. Cortés, and A. N. de Souza, "Detection and identification of abnormalities in customer consumptions in power distribution systems," IEEE Transactions on Power Delivery, vol. 26, no. 4, pp. 2436-2442, 2011.

[8] Hussain, S., Mustafa, M. W., Jumani, T. A., Baloch, S. K., Altobi, H., Khan, I. and khan, A., "A novel feature engineered-CatBoost-based supervised machine learning framework for electricity theft detection", Energy Reports, vol. 7, pp. 4425-4436, 2021. Available: 10.1016/j.egyr.2021.07.008.

[9] Hasan, M., Toma, R. N., Nahid, A. A., Islam, M., & Kim, J. M. "Electricity theft detection in smart grid systems: A CNN-LSTM based approach." Energies 12.17 (2019): 3310.

[10] Z. Zheng, Y. Yang, X. Niu, H.-N. Dai, and Y. Zhou, "Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids," IEEE Transactions on Industrial Informatics, vol. 14, no. 4, pp. 1606-1615, 2017.

[11] N. Javaid, N. Jan and M. Javed, "An adaptive synthesis to handle imbalanced big data with deep siamese network for electricity theft detection in smart grids", Journal of Parallel and Distributed Computing, vol. 153, pp. 44-52, 2021. Available: 10.1016/j.jpdc.2021.03.002.

[12] "A friendly Introduction to Siamese Networks", Medium, 2022. [Online]. Available: https://towardsdatascience.com/a-friendly-introduction-to-siamese-networks-85ab17522942. [Accessed: 07- Feb- 2022].

[13] “Siamese Networks Introduction and Implementation” [Online]. Available: https://towardsdatascience.com/siamese-networks-introduction-and-implementation-2140e3443dee. [Accessed: 02- Feb- 2022].

[14] S. K. Singh, R. Bose, and A. Joshi, "Entropy-based electricity theft detection in AMI network," IET Cyber-Physical Systems: Theory & Applications, vol. 3, no. 2, pp. 99-105, 2018.

[15] P. Jokar, N. Arianpoo, and V. C. Leung, "Electricity theft detection in AMI using customers’ consumption patterns," IEEE Transactions on Smart Grid, vol. 7, no. 1, pp. 216-226, 2015.

[16] A. Jindal, A. Dua, K. Kaur, M. Singh, N. Kumar, and S. Mishra, "Decision tree and SVM-based data analytics for theft detection in smart grid," IEEE Transactions on Industrial Informatics, vol. 12, no. 3, pp. 1005-1016, 2016.

[17] J. Pulz, R. B. Muller, F. Romero, A. Meffe, Á. F. G. Neto, and A. S. Jesus, "Fraud detection in low-voltage electricity consumers using socio-economic indicators and billing profile in smart grids," CIRED-Open Access Proceedings Journal, vol. 2017, no. 1, pp. 2300-2303, 2017.

[18] K. Wang et al., "A survey on energy internet: Architecture, approach, and emerging technologies," IEEE Systems Journal, vol. 12, no. 3, pp. 2403-2416, 2017.

[19] Arooj Arif, N. Javaid, A. Aldegheishem, and N. Alrajeh, "Big data analytics for identifying electricity theft using machine learning approaches in microgrids for smart communities," Concurrency and Computation: Practice and Experience, 2021.

[20] Y. Himeur, K. Ghanem, A. Alsalemi, F. Bensaali, and A. Amira, "Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives," Applied Energy, vol. 287, p. 116601, 2021.

[21] I. U. Khan, N. Javaid, C. J. Taylor, K. A. Gamage, and X. Ma, "Big Data Analytics for Electricity Theft Detection in Smart Grids," in 2021 IEEE Madrid PowerTech, 2021: IEEE, pp. 1-6.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778.

[23] W. Gao, A. Darvishan, M. Toghani, M. Mohammadi, O. Abedinia, and N. Ghadimi, "Different states of multi-block based forecast engine for price and load prediction," International Journal of Electrical Power & Energy Systems, vol. 104, pp. 423-435, 2019.

[24] J. Brownlee. "A Gentle Introduction to Imbalanced Classification." https://machinelearningmastery.com/what-is-imbalanced-classification/ (accessed 4 August, 2021).

[25] K. Vala. "ADASYN: Adaptive Synthetic Sampling Method for Imbalanced Data." https://towardsdatascience.com/adasyn-adaptive-synthetic-sampling-method-for-imbalanced-data-602a3673ba16 (accessed 4 August, 2021).

[26] Using Near-Miss Algorithm For Imbalanced Datasets. "https://analyticsindiamag.com/using-near-miss-algorithm-for-imbalanced-datasets/" (accessed. 4 August 2014)

[27] Y. Xiao, J. Wu, Z. Lin, and X. Zhao, "A deep learning-based multi-model ensemble method for cancer prediction," Computer methods and programs in biomedicine, vol. 153, pp. 1-9, 2018.

[28] K. Wigger. "MIT researchers warn that deep learning is approaching computational limits." https://venturebeat.com/2020/07/15/mit-researchers-warn-that-deep-learning-is-approaching-computational-limits/. (accessed 15 Dec, 2021).

[29] D. Oliva, M. Abd Elaziz, A. H. Elsheikh, and A. A. Ewees, "A review on meta-heuristics methods for estimating parameters of solar cells," Journal of Power Sources, vol. 435, p. 126683, 2019.

[30] I. U. Khan, N. Javeid, C. J. Taylor, K. A. A. Gamage and X. Ma, "A Stacked Machine and Deep Learning-based Approach for Analysing Electricity Theft in Smart Grids," in IEEE Transactions on Smart Grid, doi: 10.1109/TSG.2021.3134018.

[31] S. Blanke. "An optimization and data collection toolbox for convenient and fast prototyping of computationally expensive models." https:// github.com/ SimonBlanke/ Hyperactive(accessed 7 October 2021).

[32] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," Communications of the ACM, vol. 64, no. 3, pp. 107-115, 2021.

[33] A. Bindal. "Does Batch Norm really depends on Internal Covariate Shift for its success?" https://medium.com/techspace-usict/does-batch-norm-really-depends-on-internal-covariate-shift-for-its-success-2d854cc76838 (accessed 4 August, 2021).

[34] W. Zhang, H. Quan, and D. Srinivasan, "An improved quantile regression neural network for probabilistic load forecasting," IEEE Transactions on Smart Grid, vol. 10, no. 4, pp. 4425-4434, 2018.

[35] T. G. Slatton, "A comparison of dropout and weight decay for regularizing deep neural networks," 2014.

[36] Vu, Quang Hieu, Dymitr Ruta, and Ling Cen. "Gradient boosting decision trees for cyber security threats detection based on network events logs." 2019 IEEE International Conference on Big Data (Big Data). IEEE, 2019.

[37] B. Zeng, "Towards understanding residual neural networks," Massachusetts Institute of Technology, 2019.

[38] Electricity Theft Detection. "Electricity Theft Detection." https:// github.com/ henry RDlab/ Electricity Theft Detection (accessed 4 August, 2021).