IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Deep Neural Network-Based Electricity Theft Detection in Smart Grids

Amaresh kori
Central University of Andhra Pradesh
Anantpuramu (Andhra Pradesh)
amareshkori427@gmail.com

Dr P. Sumalatha
Assistant Prof.
Department of Artificial
Intelligence and Data Science

Abstract*—Electricity theft is a widespread issue that has a detrimental impact on both power customers and utility businesses. It impairs utility companies' ability to grow economically, creates electric risks, and has an effect on consumers' high energy costs. The development of smart grids is crucial for the identification of power theft since these systems create enormous amounts of data, including information on client use, which may be used in machine learning and deep learning methods to identify electricity theft. This research offers a method for detecting theft that use deep neural networks to classify data using extensive characteristics in the time and frequency domains. We use data interpolation and synthetic data creation techniques to overcome dataset flaws including missing data and class imbalance issues. We evaluate and contrast the influence of characteristics from perform experiments using principal component analysis in both the time and frequency domains, run experiments in combined and reduced feature space, and then apply the minimum redundancy maximum relevance approach for confirming the most crucial features. By utilizing a Bayesian optimizer to optimize hyperparameters, we may increase the performance of power theft detection. We also use an adaptive moment estimation optimizer to run tests with various values of critical parameters to find the settings that produce the greatest accuracy. Finally, we demonstrate our method's competitiveness against other approaches assessed on the same dataset. On validation, we obtained 91.8% accuracy, which is the second-best on the benchmark, and 97% area under the curve (AUC), which is 1% higher than the best AUC in existing works.

Keywords: Deep neural network, electricity theft, machine learning, minimum redundancy maximum relevance, principal component analysis, smart grids.

I. INTRODUCTION

Utility companies all across the world struggle with the issue of electricity theft. Electricity theft is the main cause of Non-Technical Losses (NTLs), which cost utility companies globally more than $96 billion annually [1]. According to the World Bank [2], 50 percent of the energy produced in sub-Saharan Africa is stolen.

The ultimate objective of energy thieves is to use energy without being charged by utility providers [3] or to pay bills that are less than the quantity of energy used [4]. As a result of power theft, utility companies lose a lot of money. According to [5], Russia lost $5.1 billion, Brazil lost $10.5 billion, and India lost $16.2 billion in 2015.

According to estimates, energy theft costs South Africa (via Eskom) some $1.31 billion (R20 billion) in lost income per year [2].

The stability and dependability of power systems are directly impacted by electricity theft, in addition to the income lost [3]. Electrical surges, overloaded electrical systems, and dangers to public safety including electric shocks can result [4]. It directly affects all customers' energy tariff increases as well [3]. The implementation of smart grids offers several chances to address the issue of power theft [4]. Traditional electricity grids, smart meters, sensors, computer capabilities to monitor and regulate grids, etc. are often included in smart grids and are all connected via a communication network [6]. Smart Data on power use, grid health, electricity pricing, and other topics are gathered through meters and sensors [6].

Many Utilities looked at how their meters were installed and configured, tested to see if the power line was bypassed, and other methods to reduce electricity theft in traditional grids [4]. These techniques are costly, ineffective, and unable to identify cyberattacks [4], [7]. Using freely available data from smart meters and machine learning classification techniques, researchers have recently attempted to identify power theft. These methods of theft detection have proven to be relatively less expensive [8]. However, the performance of existing classification techniques is constrained because they only take into account time-domain features and ignore frequency-domain features.

Even though there is significant continuing research on the detection of electricity theft, the issue still exists. The main reason for the delay in resolving this issue might be that clever While developing nations lag behind, grid deployment is taking place in developed countries [9]. The absence of connectivity infrastructure and consumers' privacy concerns regarding the data supplied by the smart meters are two obstacles to the deployment of smart grids [10]. However, according to [10], many developed and developing nations are considering using smart meters to address NTLs. According to [11], the worldwide market for smart grids will quadruple in size between 2017 and 2023, with the following main regions

dominating the development of smart grids: Asia, Europe, and North America.

In this study, we describe a technique for detecting electricity theft that is based on Deep Neural Network (DNN)-based classification and uses carefully extracted and chosen Performance of classification is improved by features alone. The State Grid Corporation of China (SGCC) developed a realistic power usage dataset, which is available at [12]. Data on power use from January 2014 to October 2016 make up the dataset. The following are the key contributions:

• Using extensive time-domain characteristics and a unique DNN classification-based algorithm, we suggest a strategy for detecting electricity theft that is based on the literature. Additionally, we suggest utilizing frequency-domain features to improve performance.

• To interpret the results and make future training easier, we use Principal Component Analysis (PCA) to perform classification with a smaller feature space and compare the results with classification performed with all input features.

• To further identify and validate the most important features, we employ the Minimum Redundancy Maximal Relevance (mRMR) scheme. Characteristics. We demonstrate that using frequency-domain characteristics rather than time-domain.

The rest of this essay is structured as follows. The literature that has been used to address the issue of power theft is covered in Section II. We briefly introduce the strategies employed in this work in Section III. Section IV describes the step-by-step process used in this study, including dataset analysis, efforts done to enhance its quality, and analysis of customer load profiles that resulted in feature extraction and categorization. We present and discuss the findings in Section V. Finally, Section VI brings the paper to a close.

II. Related Work

Many researchers have been drawn to the study of electricity theft detection in smart grids to develop techniques that reduce electricity theft. The three main types of approaches found in the literature are hardware-based, combined hardware and data-based detection methods, and data-driven methods.

Methods based on hardware [13]– [19] typically call for the installation of hardware components on power distribution lines, such as specialized microcontrollers, sensors, and circuits. These techniques are often intended to catch power theft committed by physically interfering with components of the distribution system, such as electricity meters and distribution lines. They are unable to recognize cyberattacks. power cyber-attack is a type of power theft in which the electricity meters are hacked to alter data on energy use [7].

For example, in [13], a power meter redesign was made. It made use of a microprocessor, a GSM (Global System for Mobile Communications) module, and an EEPROM (Electrically Erasable Programmable Read-Only Memory). A simulation revealed that the meter could, by-passing the meter, send a Short Message Service (SMS) if an inelegant load was attached. limited to identifying power theft committed by physically altering distribution components like electricity meters and wires. The GSM module, ARM-cortex M3 processor, and other hardware elements were used by the authors of [16] to address the issue of electricity theft, which was manifested in the four ways of bypassing the phase line, the meter, disconnecting the neutral line, and tampering with the meter to make unauthorized modifications. To test all four, a prototype was created. possibilities. For each stolen incidence, the GSM module was able to send an SMS notification.

Authors in [17] created the ADE7953 chip-based smart meter, which is sensitive to mechanical tempering in addition to current and voltage tempering. The ADE7953 was used to identify abnormalities in voltage and current, such as overvoltage, falling voltage, overcurrent, the absence of a load, and others. It informed the Microcontroller Unit (MCU) of an interrupt, which possibilities. Every theft reportable tampering status could be notified by the GSM module via SMS. By attaching a tampering switch to the IO ports of the MCU, which may transmit warning signals to the MCU when interfered with, mechanical tampering was defeated. The concept was put to the test using tampering scenarios such joining the phase and neutral lines, connecting the input and output of the meter in reverse and connecting the phase line directly to the load. The likelihood of a failed detection was 2.13%.

Authors in [15] designed a circuitry to detect power theft by comparing forward current on the main phase line with reverse current on the neutral line using a step-down transformer, voltage divider circuit, microprocessor, and other hardware components. The circuitry was already fitted ahead of the meter. Both real hardware and Proteus software were used to test the concept. The issue was discovered, and an alert went off when the meter was bypassed. [14] describes the design of a circuit to identify power theft committed by meter-bypassing. Used hardware included transformers, rectifiers, a microcontroller, a GSM module, and other items. When the meter was tampered with, the GSM controller sent an SMS notification to the operator.

The authors of [18] suggested attaching Radio Frequency Identification (RFID) tags to ammeters in order to record specific information about each ammeter. Real-time tracking and management of ammeters was required. Theft of electricity had to be investigated on site. A substantial likelihood of an electricity theft exists if a tag is damaged, removed, or has information different from the original.

Evaluation based on a study of deployment costs. Return on Investment (ROI) for a utility firm in China was determined to be more than 1. [19] describes the construction of an Arduino-based real-time power theft detector. The hardware components Arduino Uno, GSM module, current sensors, and LCD were employed. Current sensors, one on the secondary side of the transformer and the other on the electric service cap, sent measurements to the Arduino Uno. The message would be transmitted to the operator via a GSM module if the difference between the measurements of the present sensors exceeded a certain threshold. The prototype was created using hardware that, when tested, was able to notify theft incidents, and the simulation was carried out using Proteus 8 software. These techniques are costly since they need specialized hardware deployment and upkeep, which makes them ineffective at stopping cyberattacks. To address the issue of energy theft, combined hardware and data-based detection solutions [20]– [22] make use of hardware, machine learning, and/or deep learning techniques. These approaches also present the difficulty of being expensive to deploy and maintain due to hardware constraints.

[20] suggested a method for calculating a neighborhood's overall consumption and comparing the findings to the usage that neighborhood's smart meters indicated. Smart meters and

transformer measurements would diverge noticeably, which would indicate the neighborhood is home to dishonest clients. In order to identify the dishonest clients in the area, the writers.

recommended utilizing a classifier based on a Support Vector Machine (SVM). A dataset of 5000 (all devoted) customers served as the classifier's test subjects. It was possible to attain a maximum detection rate of 94% and a minimum false positive rate of 11%.

A prediction model was created by authors in [22] to calculate TLs. The entire distribution network losses would be deducted from TLs to obtain NTL. A smart meter simulator was used to produce data for 30 consumers in 30-minute intervals over the course of six days based on the premise that distribution transformers and smart meters share data to the utility after every 30 minutes. Unfaithful users on the simulator stole electricity by avoiding the meter. The amount of electricity that was stolen ranged from 1% to 10% of the overall use. stolen energy worth more than.

When one or more metres are suspected of having been tampered with, a solution was suggested in [21] that would utilize an observer metre that would be mounted on a pole far from residences and record the total quantity of power provided to n houses. To prevent tampering, cameras would be placed around the observer metre. A mathematical technique was created to identify a smart metre that has been tampered with using information from observer metres and smart metres. By boosting the consumption of some metres that were randomly selected, a mathematical approach was evaluated using a dataset of actual consumption. The system was effective in identifying metres with changed consumption.

Many researchers work in the aforementioned areas because of the high cost of demand. on data-driven solutions to the issue of power theft. For instance, the authors of [3] developed a system for detecting power theft by using three pipelined algorithms: SVM, Kernel function and Principal Component Analysis (KPCA), and Synthetic Minority Over-sampling Technique (SMOTE). In order to balance an unbalanced dataset, they employed SMOTE to create fictitious data, KPCA to extract features, and SVM to do classification. On validation, they attained the highest possible overall classifier quality, as indicated by an Area Under the Curve (AUC) of 89%.

Wide and deep Convolutional Neural Networks (CNN) models were employed by the authors of [4] to identify electricity theft. Wide was to learn numerous co-occurrences of features for 1-D series based on the fact that regular power usage is periodic whereas data on stolen electricity consumption is not periodic. although deep, data Data was aligned in a 2-D fashion by weeks and CNN was used to capture periodicity. To get a maximum AUC value of 79%, they changed the ratios of training and validation data. The strategy we provide in this research produces AUC scores exceeding 90% on both validation and testing by using the same dataset used in [3] and [4].

In [23], Principal Component Analysis (PCA) was used to extract Principal Components (PCs) from the original high-dimensional consumption data while retaining the necessary variance. The introduction of an anomaly score parameter with a defined range between lowest and maximum values. For The anomaly score parameter was determined for each test sample.

The sample would subsequently be viewed as malicious if the result did not fall within the predetermined thresholds. The method's evaluation was based on the true positive rate (TPR), which reached the highest figure ever recorded of 90.9%. One-Class SVM (O-SVM), Cost-Sensitive SVM (CS-SVM), Optimum Path Forest (OPF), and C4.5 tree were utilized by the authors in [24]. distinct characteristics were chosen from consumer consumption data, and the performance of each classifier was examined independently on a distinct set of features. All classifiers were then combined for the best outcomes. With 86.2% accuracy, the best results were obtained when all classifiers were merged.

Long Short-Term Memory (LSTM) recurrent neural networks and CNNs were used in conjunction by the authors of [25]. We employed hidden layers, of which CNN and LSTM each used four and three, respectively. This approach depended on CNN's capability to automatically extract features from a given dataset. 1-D time-series data were used to extract features. The highest accuracy on model validation was 89%. To identify energy theft, the authors in [26] used the Local Outlier Factor (LOF) with k-means clustering. Customers' load proles were analyzed using k-means clustering, and customers whose load proles were far from their cluster centers had their anomaly degrees calculated using LOF. They were able to evaluate the approach with an AUC of 81.5%. Our model has a maximum accuracy score of 91.8% and a validation score of 97%.

[27] Two energy theft incidents There were created models. The first model is based on the classier Light Gradient Boosting (LGB). The dataset was balanced using a mix of SMOTE and Edited Nearest Neighbor (ENN). Alex Net was used for feature extraction, and LGB for classification. The suggested model was given the acronym SALM (SMOTEENN-Alex Net-LGB). The second model is more sophisticated and is based on adaptive boosting. To balance the data of the imbalanced classes, synthetic data that approximated the minority class data was produced using the Conditional Wasserstein Generative Adversarial Network with Gradient Penalty (CWGAN-GP). Utilizing Google Net for feature extraction, classification by AdaBoost was done next. The suggested model was given the moniker GAN-NET Boost. The SGCC data utilized in this work were used to assess the models. 90% accuracy was reached with SALM and GAN-Net Boost, and 95%, and an AUC on validation of 90.6% and 96%, respectively.

Although these models were able to provide excellent results, their performance was constrained by the fact that they only took time-domain data into account. Our technique demonstrates that performance of classification is enhanced by adding frequency-domain characteristics on top of time-domain features.

III PRELIMINARIES

The three primary methods—Deep Neural Networks (DNNs), Principal Component Analysis (PCA), and Minimum Redundancy Maximum Relevance (mRMR)—are summarised in this section.

A. Deep neural networks

Artificial neural networks (ANNs) are a family of machine learning methods created to mimic the biological workings of the human brain [28, 29]. They are frequently applied to extract patterns or spot trends that are challenging to identify using other machine learning approaches [30].

They are made up of several layers of nodes or neurons coupled to further layers [29]. The fundamental building block of a neural network is a neuron, which derives from the McCulloch-Pitts neuron, a simplified model of a neuron in the

human brain [31]. The model diagram of a neuron is shown in Figure 1. that includes a layer that comes after the ANN's input.

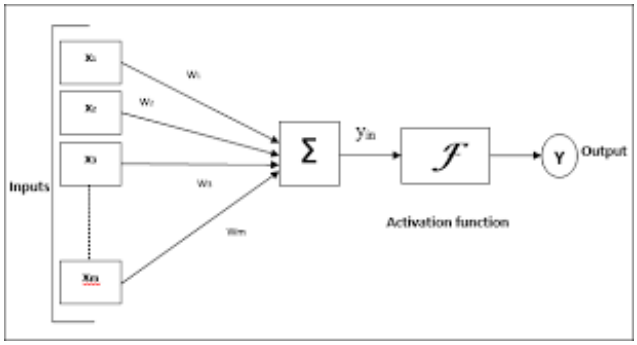


FIGURE 1. First hidden layer neuron model.

Research on ANNs is where the idea for Deep Neural Networks (DNNs) first emerged [32]. Two or more hidden layers are a defining characteristic of DNNs [28]. Compared to shallow ANNs, they can learn characteristics that are more complicated and abstract [33]. The output layer of classification problems is frequently constructed so that one neuron stands in for each class [29].

All levels of the neural network, with the exception of the output layer that classifies using the learned features, are employed to filter and learn the complex features [29], [34]. Prior to the creation of DNNs, the majority of machine learning approaches investigated shallow structural topologies, which typically include just one layer of non-linear transformation [32]. SVMs, logistic regression, and ANNs with a single hidden layer are a few examples of these designs.

DNNs use a variety of architectures, such are employed to address various issues. Convolutional, recurrent, and feed-forward DNN architectures are a few examples of DNN architectures.

A fully linked feed-forward model is used in this research effort.

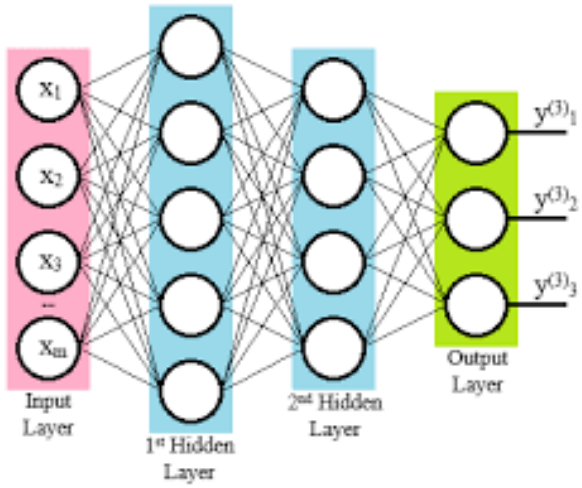


FIGURE 2. Fully connected feed-forward DNN general architecture.

It was DNN. Figure 2 depicts the standard layout of a fully linked feed forward DNN.

The following are the main components of the DNN shown in Figure:

- Layer that contains the characteristics or representation of input data (x).
- Weights of the connections between a DNN's input layer and first hidden layer are known as input weights (wi).
- concealed layers
  the neuronal layers that lie between the input and output layers. They are applied to investigate how the input and output signals interact [30].
- Weights of hidden neurons ([wh1…….whk])
  linkages between the unseen layers that weigh more.
- Weights of the output (wo)
  weights between the output layer and the final hidden layer.
- The output layer (y) is DNN's last layer. It provides the results of the

Computation in a feed-forward architecture is a series of operations performed on a preceding layer's output. The output is produced by the last procedures. The output for a particular input remains constant; it is independent of the prior network input [33].

1) DNNs' HISTORY OF DEVELOPMENT

According to [33], research on DNNs began in the 1960s, whereas ANNs were first suggested in the 1940s. For reading handwritten numbers, the LeNet network, which made use of several digital neurons, was developed in 1989. Beyond 2010, significant advancements were made, with examples being Alex Net's picture recognition and Microsoft's speech recognition systems.

recognition system, as well as DNN accelerator research from Neuron and DianNao.

According to [30], [32], and [33], the following factors have been crucial to the development of DNNs:

improvements in computer design and semiconductor technology that enable parallel computation and decrease hardware prices.

Large datasets are created by the enormous quantity of data that cloud providers and other companies collect, successfully training DNNs.

Research advancements in signal/information processing and machine learning have led to the development of strategies to increase accuracy and widen the application domain for DNNs.

DNNs can contain more than a thousand layers, as long as the technology allows it [33].

2) DNN TRAINING

The two main prerequisites for training the DNN are a huge dataset and strong computing capabilities. given that weight adjustments demand several iterations [33]. The weights between the neurons are adjusted during the DNN training process [30].

The DNN gains knowledge from the data during the training phase. There are four main ways to learn: supervised, semi-supervised, unsupervised, or by reinforcement [33][36].

Supervised learning was applied in this study. According to [28], [34], the standard process for supervised learning in DNNs is as follows:

- Appropriate starting values are used to initialize the weights W D [wi; wh1…..whk;wo].
- The input layer of the network receives input signal x.

- In order to decrease error, output error is determined and weights are then changed.
- For all training data, repeat steps 2 and 3.

3) BACKPROPAGATION

Weights from subsequent layers between the input and output layers make up the loss function of a multi-layered ANN [36]. Chain rule is used in backpropagation to calculate the gradient of the loss function as the sum of local gradient products over various node connections between input and output layers [28], [29], [36]. Backpropagation methods often adjust the neural network parameters on each layer using gradient-based optimisation techniques [37].

4) ACTIVATION FUNCTION

By replicating the action of a biological neuron, an activation function converts an input signal into an output signal that might be an input to another neuron [38, 39]. There are several activation functions, and they may be broadly categorized into two categories: linear activation functions and non-linear activation functions. The form

$$g = f(z) \quad (2)$$

5) FUNCTIONS FOR LINEAR ACTIVATION

Typically, the activation of a linear activation function is directly proportional to the input. Equation (3) can be used to represent them.

$$f(z) = c(z) \quad (3)$$

C is a constant

The derivative of the linear activation function is f 0(z) D C, and its output falls within the range (1;1). The application of a gradient cannot reduce an error since it is unrelated to the input [40]. Regression issues often employ this activation function [41].

6) NON-LINEAR FUNCTIONS FOR ACTIVATION

Because they can adjust to changes in the input and distinguish between outputs, non-linear activation functions are frequently utilised in DNNs [40]. The most well-liked non-linear activation functions among the numerous that have been created are

- *Sigmoid activation function*

Sigmoid activation function is given by Equation

Sigmoid / Logistic

$$f(x) = \frac{1}{1 + e^{-x}}$$

B Principle component Analysis

From a data table of inter-correlated features/variables that represent observations, PCA [42] is used to extract crucial information. Principal Components (PCs), a new collection of orthogonal variables, are used to represent this extracted data. The Singular Value Decomposition (SVD) method [43], which is used in this study, operates in the following way: The SVD divides the input feature matrix X into three matrices, namely, X D PQRt,

- **P** is the normalized eigen vectors of the matrix $\mathbf{XX}^T$,
- _ **Q** D **E**1
- 2 where **E** is a diagonal matrix of eigen values of matrix $\mathbf{XX}^T$, and
- _ **R** is the normalized eigen vectors of matrix $\mathbf{X}^T\mathbf{X}$

so that V D1 are produced and are sorted in decreasing order of variance [23]. A PC is given by for position p.

IV DNN-BASED ELECTRICITY THEFT DETECTION METHOD

The three phases that make up the approach for detecting electricity theft in this section are as follows: Feature Extraction, Data Analysis and Pre-processing.

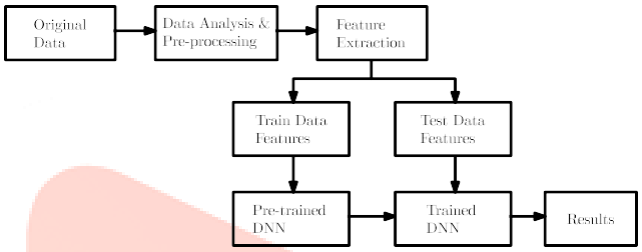


FIGURE 3. Electricity theft detection workflow diagram.

A. DATA ANALYSIS AND PRE-PROCESSING

We describe the dataset used in this subsection and how we improved its quality by locating and eliminating occurrences without consumption data. An observation in this study refers to a single instance or record from the dataset that was used for the duration of consumption measurement. I.e., given dataset A of size N, ai A, where ai is the ith observation of A and 1 i N.

We display a study of consumer load profiles. We also provide information on data interpolation and the creation of fake data.

1) DATASET ANALYSIS AND PREPARATION

As mentioned in Section I, we used a real-world power consumption dataset made available by SGCC and found at [12]. The dataset is made up of daily power use information collected between January 2014 and October 2016, which is presented in Table 1. Every client receives the same sample rate of one measurement each day, which corresponds to their daily average power usage. The utilised dataset has 42372 observations total, 3615 of which are power consumption data of dishonest customers while the remaining observations are electricity consumption data of loyal customers.

TABLE 1. Dataset summary table.

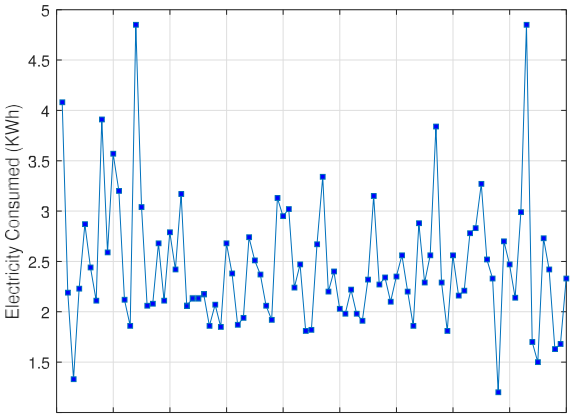
Number of observation days: 1,034

Customer Class	Number Of Observations		
	From Original Dataset	*After Removing Empty Observations*	*After Synthetic Data Generation*
Faithful	38,757	36,679	36,679
Unfaithful	3,615	3,579	36,679
Total	**42,372**	**40,258**	**73,358**

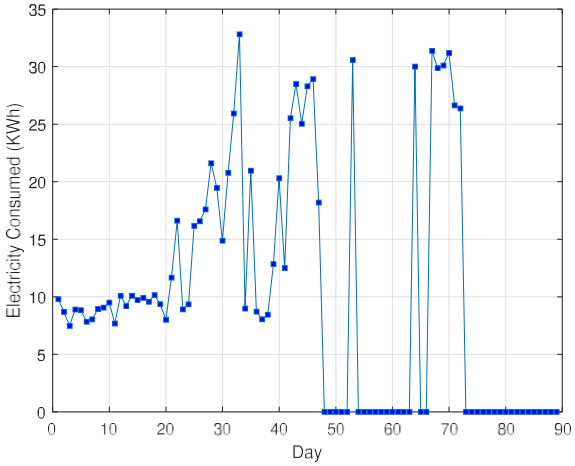
Data, like many datasets used in the literature, has a significant amount of inaccuracies brought on by difficulties with smart metres, data storage, data transfer, and unplanned system maintenance, among other things [4]. This study's dataset is not an exception. It is made up of remnants of non-numerical or empty values. Using data analysis techniques, we discovered that throughout the whole 1034-day period, around 5.45% of the observations in this dataset were zeroes, just null values, or a mix of the two. These observations were thought to be meaningless. Specifically, if ai 0 or ai / R for any ai a, an observation is said to be empty. Since these observations do not contain any properties that distinguish the classes, they cannot be used to any electricity consumption report that is more than 0 kWh. These observations were eliminated in order to enhance the dataset's quality. They were rejected because, while being labelled with either class, they could not be assigned to either class. Table 1's third column displays a list of the observations that are still present after empty observations have been eliminated.

Figure 4 displays line plots of three-month consumption data for a faithful client and an unfaithful customer versus consumption days. In contrast to the consumption behaviour of the electrical thief, which takes on various shapes and is unpredictable, the consumption behaviour of the honest client is largely uniform and has a predictable trajectory. We also conducted histogram analysis for the two classes. of clients, as seen in Figure 5.

We can see from the shown histograms that, when compared to dishonest consumer consumption statistics, statistical metrics mean, mode, and median are often closer to the histogram centre. We conducted a similar study for several clients and discovered that the observation made here holds true for the majority of the dataset. Based on these findings, we claim that by designating outliers



(a) Faithful customer's consumption plot



(b) Unfaithful customer's consumption plot

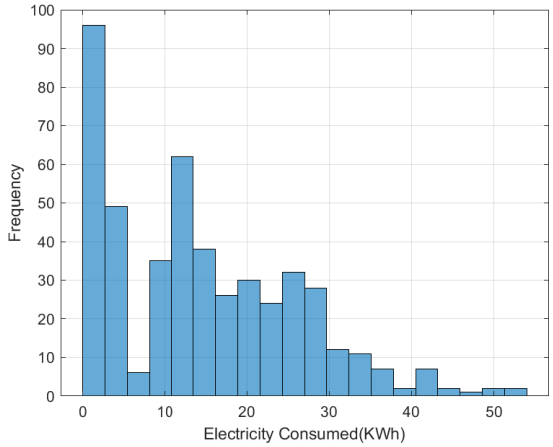
FIGURE 4. Faithful and unfaithful customers' consumption plots.

as values beyond three Median Absolute Deviations (MAD), honest customers can be characterized as having fewer outliers percentage in a given data, than unfaithful customers.

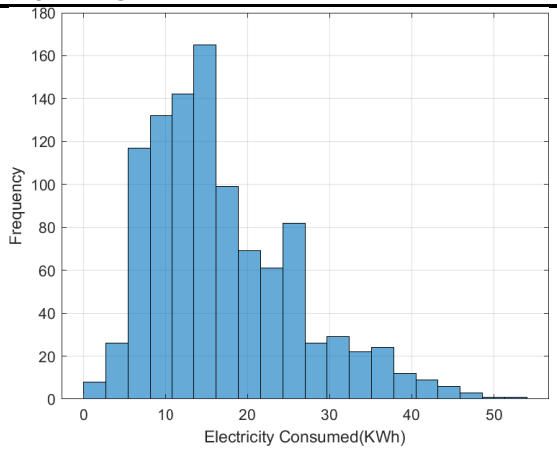
1) DATA INTERPOLATION

Data were interpolated for all observations including a mix of null or non-numerical values and actual consumption values. In order to preserve consumption patterns during data interpolation, piecewise Cubic Hermite Interpolating Polynomial (PCHIP) [46] was employed to fill in missing data.

A cubic Hermite interpolating polynomial H (x) is a shape-preserving interpolant that applies to a sub-interval of xi x xi 1 and maintains data monotonicity. The raw data mean was assessed without accounting for NaN values and then put as the first vector element for the data consumption vector that had NaN values at the beginning. PCHIP was used to complete the remaining components. This kept usage in check and stopped outliers from being added to the data.

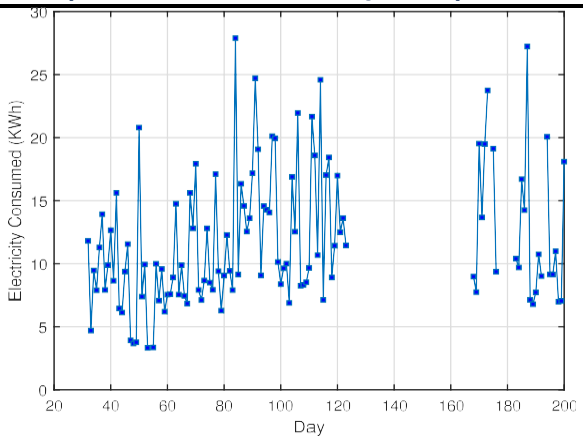


(a) Faithful customer's consumption histogram



(b) Unfaithful customer's consumption histogram

FIGURE 5. Faithful and unfaithful customers' consumption histograms.



(a) Consumption data before interpolation

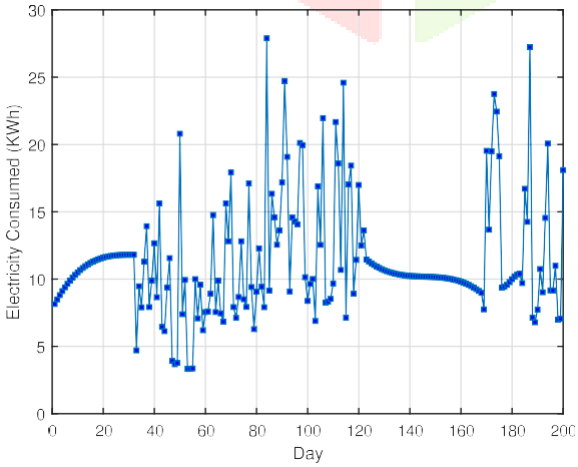
(b) Consumption data after interpolation

FIGURE 6. Plots of consumption data before and after interpolation

Figure 6 depicts a sample of one observation that was taken at random and then interpolated. For the sake of clarity, a consumption period of 200 days surrounding days with incomplete consumption data is displayed. As illustrated in Figure 6b, interpolated data points form a smooth curve that sits between the lowest and maximum near points without overshooting. The consumption data is protected in this way from the insertion of outliers and data points that may cause the interpolated data pattern to match the dishonest customer's consumption pattern of the minority class of unfaithful customers, as illustrated in Figure 4b.

1) SYNTHETIC DATA GENERATION

We performed the initial classification after removing empty observations and interpolating data. Using the dataset as-is for testing, we found that the classifier satisfactorily identified loyal customers but struggled to identify dishonest ones because of a class imbalance issue [20], [25]. A class imbalance problem arises when there are significantly more observations in one class than in another.



the quantity of observations made in the opposite class. In a class-balanced task, classification models accurately classify the majority class on a dataset while misclassifying the minority class [25]. In the dataset utilised for this study, the number of faithful customers is significantly higher than the number of unfaithful customers.

This method of producing synthetic data is quick and inexpensive to implement since it makes use of the class of faithful customers' existing data to produce data for the opposite class. It just requires multiplying the observed data by a matrix of randomly generated integers, which is a single action on the recorded data. The obtained data was added to the initial dataset and assigned the consumption class of dishonest consumers. A overview of the observations made following the development of synthetic data is included in Table 1's fourth column.

B. FEATURE EXTRACTION

Electricity consumption data used in this project is *univariate* time-series data. A univariate measurement is a single measurement frequently taken over time [47]. For solving classification problems, data can be represented by its features (properties), which can then be fed as input to the classifier, as is the case in [29], [34] and [48]. Data is classtifled based on the similarity between features [47] given a dataset of different samples. In this work, time-domain and frequency-domain features were extracted and used as input to a deep neural network for classification. Classification performance comparison between time-domain, frequency-domain and combined features from both domains was carried out.

1) TIME-DOMAIN FEATURE EXTRACTION

According to line plots and histogram graphs, faithful and unfaithful consumers' consumption data clearly differentiates by a pattern of consumption, as seen in IV-A. Based on this knowledge, the time-domain traits listed in Table 2 may be utilised collectively to distinguish between the two client segments. Aside from the finding that the consumption statistics of loyal customers generally follow a pattern.

TABLE 2. Time-domain and frequency-domain features table.

Time-domain features	Frequency-domain features
Standards probability (stdsProb)	Harmonic frequency (hfInd)
Standards mean (stdsMean)	Harmonic frequency amplitude (hfAmp)
Standards standard deviation (stdsDev)	99% spectrum bandwidth (bww99)
Outliers probability (outsProb)	Lower bound frequency (flb)
Outliers mean (outsMean)	Upper bound frequency (fub)
Outliers standard deviation (outsDev)	99% bandwidth power (bwwpwr)
Data mean (dataMean)	50% bandwidth (bw50)
Data mode (dataMode)	Median frequency (fmed)
Data median (dataMedian)	Mean frequency (fmean)
Average of pchip interpolant curve fitted parameters (cfpMean)	

Unfaithful consumers' consumption patterns are not predictable; as seen in Figure 4, they do not use the same quantity of energy throughout any given period of time. Due to several factors, including the number of appliances utilised, the kind of appliances in each household, the size of the household, etc., each customer's energy consumption may vary. All observations are made to fit inside the same axes in order to increase the classification accuracy. This is accomplished by utilising the Min-Max approach [49] from Equation (17) to normalise the data for each observation. The Min-max approach maintains the initial consumption pattern while shrinking the data between 0 and 1.

$$f(x_i) = \frac{x_i - min(\mathbf{x})}{max(\mathbf{x}) - min(\mathbf{x})}$$

1) FREQUENCY-DOMAIN FEATURE EXTRACTION

According to the Fourier theorem, a periodic signal x(t) may be represented as the sum of complex sinusoidal signals whose frequencies are integer multiples of the fundamental frequency fT [50]. The consumption data graphs in IV-A may be seen as a time series signal that can be translated into the frequency-domain using the Fourier transform, according to the Fourier theorem. We retrieved frequency-domain characteristics from each observation that were represented in the frequency-domain. Using Equation (17), features were normalised after being extracted so that they could be provided as input to the classifier since neural networks are sensitive to a variety of input data. Feature extracts from both domains are displayed in Table 2.

A. CLASSIFICATION

1) NETWORK ARCHITECTURE

A fully connected feed-forward DNN architecture shown in Figure 7 was used for the classification process.

In order to avoid network underfitting and overfitting [35], the following rule of thumb methods [35], [51] were considered in the design of hidden layers of a deep neural network classifier shown in Figure 7:.

- Number of hidden neurons should be between the size of the input layer and size of the output layer,
- Number of hidden neurons should be approximated to the summation of $\frac{2}{3}$ size of input layer and size of the output layer.
- Number of hidden neurons should be less than twice the size of the input layer.

Rectified Linear unit (ReLU) activation function was used in the hidden neurons because of its better convergence property in comparison to other activation functions [28].

2) TRAINING

The classification approach was divided into four parts, the first of which used only time-domain features for classification, the second of which used only frequency-domain features, the third of which included combined features from both domains, and the fourth of which used PCA to perform classification in a smaller feature space. The maximum number of training iterations was set at 1000.

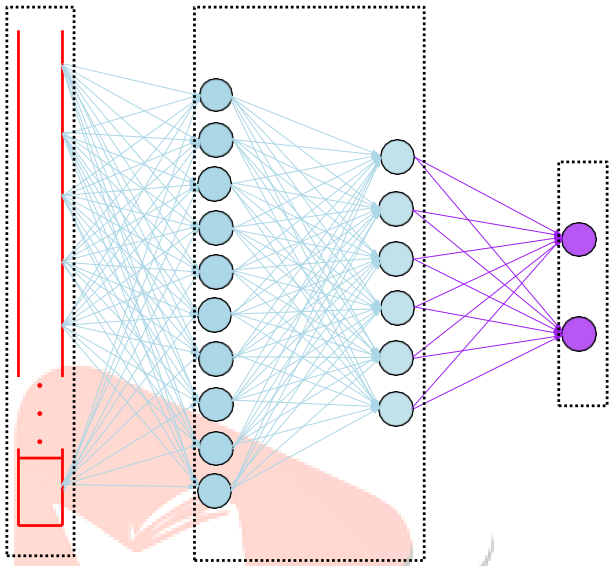


FIGURE 7. DNN classifier architecture.

Table 1. Font sizes for papers. Table caption with more than one line must be Title.

As a general rule, 80% of the total data was used for training and validation in all methods, and 20% of the total data was utilised for testing. This is known as the holdout validation scheme. 80% of the training data and 20% of the validation data were taken from the training set. When employing the k-fold cross-validation procedure with k 5, similar results were achieved. [52] provides an example of how to use a k-fold cross-validation strategy with k 5.

3) PERFORMANCE METRICS

We evaluated the performance of the classifier using the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) derived from a confusion matrix [41]. Precision/Positive Predictive Value (PPV), F1-Score [55], Matthews Correlation Coefficient (MCC) [25], Accuracy and Area Under the Curve of Receiver Operator Characteristic (AUC-ROC) curve [56] are some of the metrics used to measure recall and true positive rates. We briefly introduce the following performance measurements.

The percentage of properly labelled positive instances, or recall or true positive rate (TPR), is measured.

4) HYPERPARAMETERS OPTIMIZATION

We tuned the following hyperparameters using the Bayesian optimisation approach [57] to get the best classification performance in a reasonable amount of time: the number of hidden layers, the size of each layer, the regularisation strength, and the activation function. The Bayes theorem, which stipulates that for occurrences A and B, the probability of each.

$$P(A|B)=\frac{P(B|A)P(A)}{P(B)}.$$

By assuming that an optimisation function follows the Gaussian distribution, this optimisation technique determines the distribution of hyperparameters. One hundred optimisation steps were taken to find the ideal set of hyperparameters. Similar to the network in Figure 7, the resulting optimised network was trained and evaluated.

5) IMPACT OF KEY PARAMETERS INVESTIGATION

Initial learning rate, mini- batch size, and l2-regularization parameter effects on the optimised network were investigated using adaptive moment estimation (Adam) optimizer [58]. Both training and validation data were separated into separate categories.

The amount of training, validation, and test data is crucial to the success of the categorization. Less data is required for training the more strongly the input characteristics correlate with the class label [59]. However, it is not advised to use less than 50% of a dataset as training data because the test results would suffer [59]. In light of this, we evaluated the influence of factors using various training data percentages.

TABLE 3. Investigated parameters table.

	Initial Value	Log Step	Final Value	Fixed Value
Initial learning rate	10^{-5}	$10^{0.03}$	10^{-2}	10^{-4}
Minibatch size	10^{1}	$10^{0.04}$	10^{5}	128
L2-regularization	10^{-8}	$10^{0.06}$	10^{-2}	10^{-5}

We carried out the following procedure for 60%, 70% and 80% training data portions. For each parameter, its impact was investigated by determining training and validation accuracies with varied parameter values. Parameters were logarithmically varied in 100 steps between the initial and final values. For each step, the number of training epochs was limited to 30. The other parameters were held at fixed values while adjusting a parameter under study. Table 3 shows investigated parameters' initial values, step values, final values as well as fixed values.

V. RESULTS AND DISCUSSION

We provide and discuss the experimental findings in this section. Results that were acquired prior to the creation of synthetic data are presented in Section V-A. In Section V-B, we compare classification performance when employing characteristics from the frequency-domain, time-domain, and both domains together as inputs to the classifier. In Section V-C, we analyse the effect of PCA dimensionality reduction on experimental outcomes. In Section V-D, we give the best Bayesian optimisation findings in addition to the results obtained using optimised classifiers. In Section V-E, we give a study of the ideal parameter settings for the best classification performance by adjusting various parameters using the Adam optimizer. Finally, in Section V-F, we compare our technique to data-based power theft detection systems that have been developed in the literature.

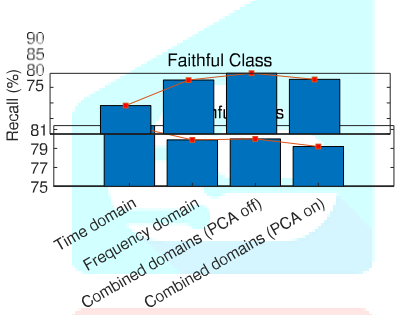
A. *VALIDATION RESULTS BEFORE SYNTHETIC DATA GENERATION*

As was said in Section IV, the classifier did poorly on the class with the significantly less observations when there was an imbalance in the number of observations between the two classes. The classifier displayed in Figure 7 was trained using characteristics taken from an actual dataset without the addition of any additional synthetic data. 20% of the data was utilised for validation, while the remaining 80% was used for training. The validation findings are displayed in Table 4's third column. Results of validation are substantially better for the faithful customers class than for the unfaithful customers class. This is demonstrated by comparing the recall, precision, and F1- score between loyal and *disloyal* consumers.

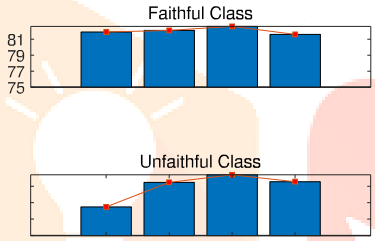
There was no improvement in validation outcomes in combined domains prior to PCA. Since the difference in the corresponding values was within a 1% margin, there was a substantial shift in the recall, accuracy, and F1-score for the class of devoted customers. But for the dishonest class

Parameter	Class	Before synthetic data generation	After synthetic data generation							
			Time-domain		Frequency-domain		Combined Domains			
							PCA Not Used		*PCA Used*	
		Val(%)	Val(%)	Test (%)	Val(%)	Test (%)	Val(%)	Test (%)	Val(%)	Test (%)
Recall	Faithful	94.6	85.8	84.1	92.8	92.3	94.2	94.5	93.0	92.5
	Unfaithful	4.3	89.2	81.4	90.4	79.9	90.0	80.0	89.0	79.2
Precision	Faithful	91.4	88.8	81.9	90.6	82.1	90.4	82.6	89.4	81.6
	Unfaithful	6.9	86.3	83.7	92.7	91.2	93.9	93.5	92.7	91.4
F1-Score	Faithful	93.0	87.3	83.0	91.7	86.9	92.3	88.2	91.2	86.7
	Unfaithful	5.3	87.7	82.5	91.5	85.2	91.9	86.2	90.8	84.9
Accuracy		86.9	87.5	82.8	89.9	86.1	91.1	87.3	90.5	85.8
AUC-ROC		66	94	90	96	92	97	93	96	92

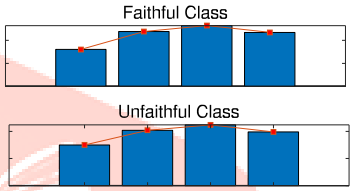
MCC = 0.84 (on validation) and 0.75 (on test).



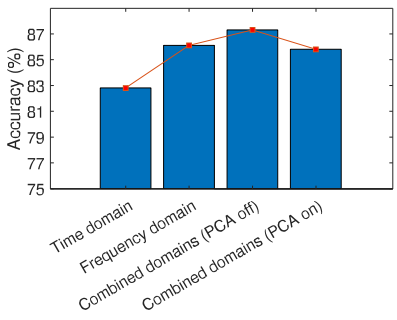
(a) Recall



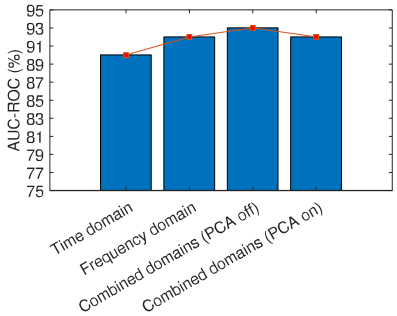
(b) Precision



(c) F1-Score



(d) Accuracy



(e) AUC-ROC

FIGURE 8. Performance metrics graphs.

Prior to balancing the classes, validation results for the minority class, which measured recall, accuracy, and F1-score, were quite poor. After balancing the classes, a noticeable improvement was made. This demonstrates that the classifier's sensitivity to the minority class was inferior to its sensitivity to the dominant class.

The findings that were achieved after adding synthetic data to the original dataset to balance classes are shown in the following subsections.

B. DIFFERENT DOMAINS FEATURES' CONTRIBUTION ANALYSIS

We give experimental findings based on well recognised performance indicators, which are compiled in Table 4, to demonstrate the dependability and robustness of the technology introduced in this study. Analysis is made easier by comparing categorization accuracy between Figure 8 shows a visual representation of time-domain, frequency-domain, and integrated information from both domains.

As can be seen from Table 4 and Figure 8, the classification technique when combined with time-domain characteristics produced remarkable validation and test results for both groups of faithful and unfaithful customers. An experiment using only frequency-domain characteristics produced better outcomes. When all of the characteristics from both domains were integrated, the best results were attained. For instance, when the experiment was conducted using time-domain features, frequency-domain features, and all characteristics from both domains, respectively, accuracy on validation was 87.5%, then improved to 89.9%, and eventually increased to 91.1%. The red trend line in Figure 8 graphs shows a notable improvement in the outcomes of tests using time-domain and frequency-domain characteristics.

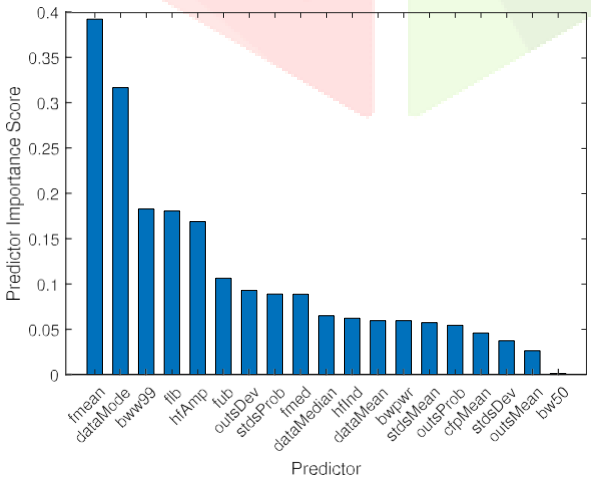


FIGURE 9. Features presented in order of their prominence.

Figure 9 is a bar chart created by the mRMR scheme that displays the predictors in order of their significance.

The bar chart in Figure 9 demonstrates that more frequency-domain variables—those with the highest scores—are located to the left of the bar chart than time-domain data, with mean frequency receiving the greatest predictor score. By performing classification tasks utilising the top 3, middle 3, and bottom 3 features on the same network in Figure 7, we validated the accuracy of the features' ranking using the mRMR scheme. The classification accuracy and AUC-ROC findings are displayed in Figure 10's bar graph.

Figure 10's findings were compared, and we found that accuracy and AUC-ROC performed best for the top three characteristics and worst for the bottom three. as anticipated features. When all characteristics were merged in the previous experiment, MCC was calculated. On validation and test, it had values of 0.84 and 0.75, respectively, which are nearer to 1 than 0. On the validation and test runs, the AUC-ROC values were found to be 97% and 93%, respectively. These outcomes show a successful categorization task in its entirety.

C...ANALYSIS OF COMPONENTS REDUCTION WITH PCA

Seven components were still present after PCA was used using the component reduction criterion of leaving enough components to account for 95% of the variance.

Contributions to the overall variation in percentage terms were made by 35.84%, 27.02%, 15.55%, 7.69%, 4.87%, 3.30%, and 1.81%. Figure 11 displays 2-D biplots of the original feature contributions to each of the primary component space components.

Time-domain characteristics are denoted with a 't' suffix, whereas frequency-domain features are denoted with an's'. The vector direction and length of each feature indicate how much that feature contributes to the major component. Figure 11 shows that frequency-domain characteristics contributed more to the major components than time-domain features. This was further supported by a study of the features importance scores presented by

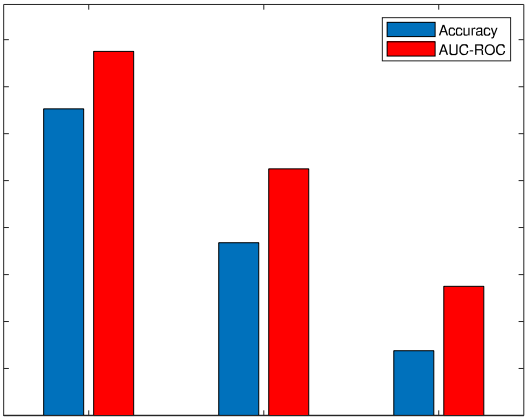


FIGURE 10. Classification results comparison of features ordered by mRMR scheme.

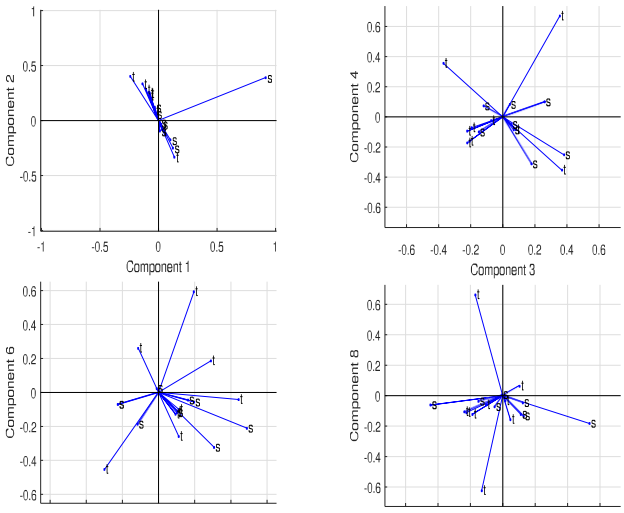


FIGURE 11. Graphical display of original features' contribution to principal components.

Based on the mRMR system, Figure 9. Table 4's last two columns display the test and validation findings that were attained after the components were reduced using PCA. We found that by using only seven main components, we could attain outcomes that were almost identical to those obtained when no feature reduction criterion was applied.

D. HYPERPARAMETERS OPTIMIZATION RESULTS

In accordance with the hyperparameters optimisation process outlined in Section IV-C4, Figure 12 plots the observed values of the objective function against the various stages of optimisation.

The optimal combination of hyperparameters was discovered at the 26th optimisation phase and stayed constant through the 100th step. Table 5 displays their values. Maximum validation and test accuracies of 91.8% and 88.1%, respectively, were attained using an enhanced classification network architecture built with optimised hyperparameters, which is 0.7% and 0.8% higher than an unoptimized design.

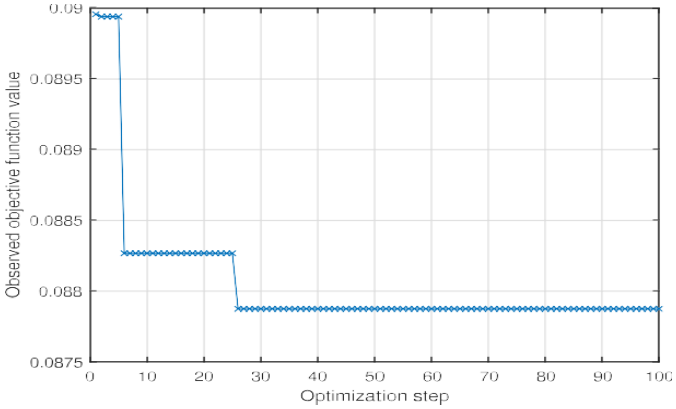


FIGURE 12. Objective function value vs optimization steps

TABLE 5. Optimized hyperparameter values.

Parameter	Value
Fully-connected Layers	[41 21]
Regularization strength	5.6882×10^{-7}
Activation function	Sigmoid

architecture. The classifier's AUC-ROC score peaked at 97%.

A. KEY PARAMETERS' IMPACT ANALYSIS

1) IMPACT OF INITIAL LEARNING RATE

The starting learning rate was changed between 105 and 102 in 100 steps to examine the effects it had on training and validation accuracies. Figure 13 displays findings scatter plots with fitted curves to streamline analysis.

The lowest initial learning rates for all examined training data sections resulted in training and validation accuracy values with reported values less than 90%. For initial learning rate values between 105 and 104, there was a noticeable improvement in both accuracies. Low levels of accuracy were achieved in this range because greater training iterations and longer training times are needed for models with lower learning rates to converge to satisfactory results. As a consequence, accuracy was primarily constrained by the number of epochs permitted. the network to be trained. Average training and validation accuracies increased over 90% after the initial learning rate of 104. The best accuracy values were found for initial learning rates in the range [103.7, 102.5] for all training data segments. As the initial learning rate got close to 102, both accuracies began to decline. An initial learning rate in the range [103.7, 102.5] is advised for optimum accuracy in order to achieve the best outcomes.

2) IMPACT OF MINIBATCH SIZE

The minibatch size was changed between 101 and 105 in 100 steps to assess the effect it had on accuracy. Accuracy vs training and validation data is presented.

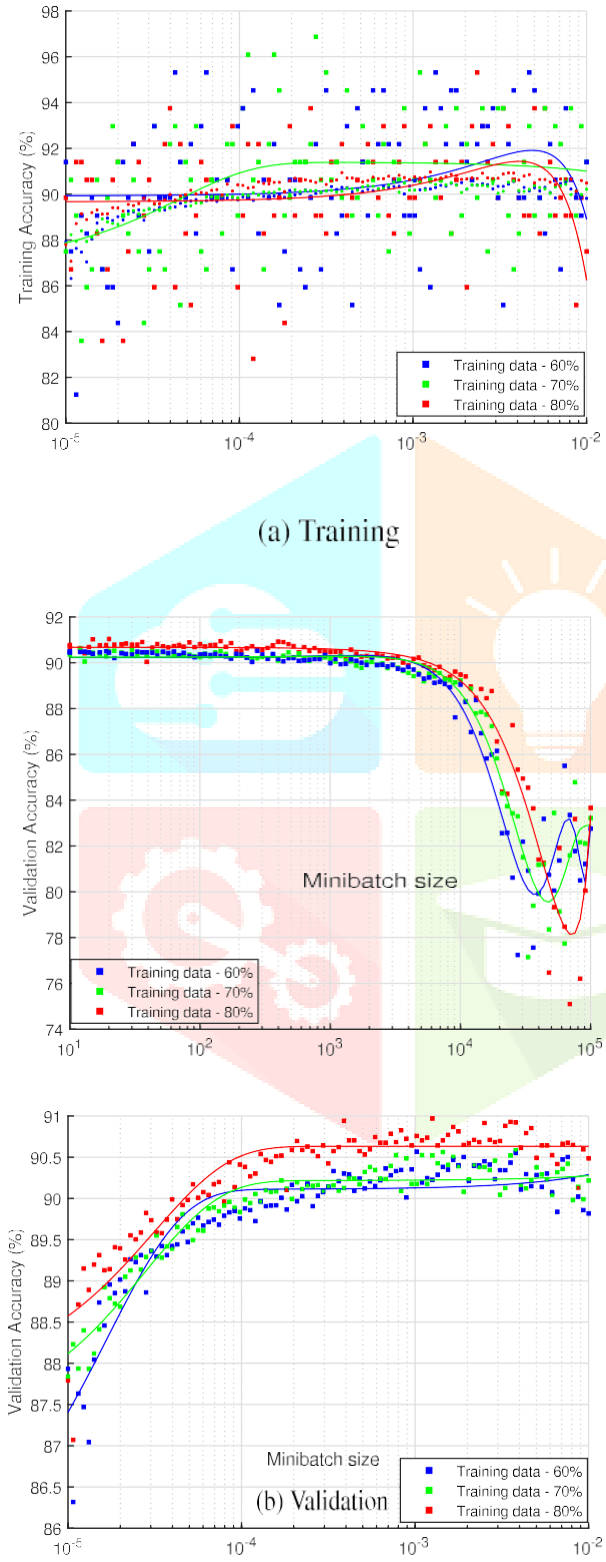


FIGURE 13. Impact of varying initial learning rate on accuracy at different training ratios.

Figure 14 shows the minibatch size parameter charts. The training and validation accuracy averages for all examined training data sections were marginally greater than 90% for minibatch size values below 103. The training accuracy fluctuated dramatically between 80% and 100% for each training task for minibatch sizes closer to 101, however this had no effect on validation because the validation accuracy remained constant at little about 90%. As minibatch size exceeded 104, both training and validation accuracy rapidly decreased. This is so that the model could learn from larger amounts of data as the value of the minibatch size grew, which led to poor generalisation. However, training a model took a long time for smaller minibatch size values. a smaller minibatch size.

3) IMPACT OF L2-REGULARIZATION PARAMETER

For the purpose of evaluating the L2-regularization parameter's effect on validation accuracy,

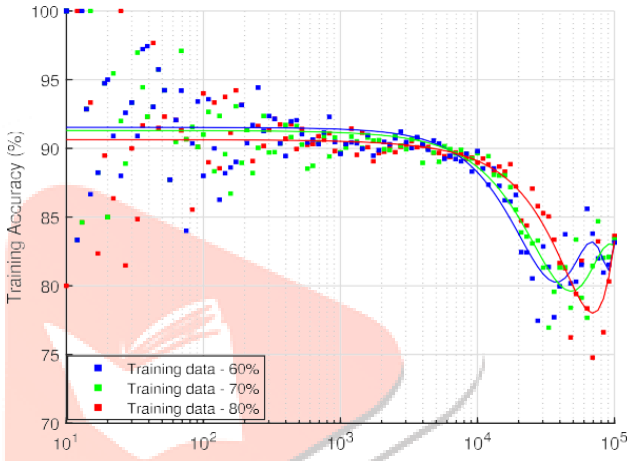


FIGURE 15. Impact of varying l2-regularization parameter on accuracy at different training ratios.

the model, the more effective it is in spotting electricity theft.

F. COMPARISON WITH EXISTING DATA-BASED ELECTRICITY THEFT DETECTION METHODS

Different data-driven approaches have been employed to address the issue of power theft based on information about electricity users' consumption. Many approaches have been tested on various unusual datasets due to the dearth of datasets comprising consumption data from both loyal and dishonest clients. We analyse the differences between our study and recent literature in Table 6 and show them. Details about the dataset are provided for each work. We examine the approaches and/or algorithms applied as well as the characteristics that were taken from the data in each method.

We compare the findings in terms of AUC and accuracy percentages for the four approaches that employed the same dataset as ours (References [3], [4], and [27]). We obtained an AUC.

Initial learning rate

contributed to the classification job and used the mRMR method to demonstrate the superiority of frequency-domain.

optimise hyperparameters, which saw an improvement in accuracy of about 1%, during validation. The Adam optimizer was implemented, and the best values for important parameters were investigated.

We attained 97% AUC, which is 1% better than the best AUC

TABLE 6. Comparison with existing data-based electricity theft detection m

Reference	Techniques/ Algorithms Used	Features used	Evaluation Dataset Details		Performance Evaluation
			Source	# of Customers	
[3]	SMOTE + KPCA + SVM.	Extracted from the original time-series data with KPCA.	SGCC	42372	Accuracy: 89% Precision: 85% Recall: 88%
[4]	Wide + deep CNN.	Wide and deep CNN used to learn features from the time-series data.	SGCC	42372	AUC: 79% Mean Average Precision (MAP): 96.9%
[23]	PCA + Calculation of anomaly score	PCs extracted from the original time-series data using PCA.	Irish smart meter data	5000	TPR: 90.9%
[24]	O-SVM + CS-SVM + OPF + C4.5 tree	Manually selected features from the original time-series data	Uruguayan electric power company	1504 3338 (two independent datasets)	Best accuracy: 86.2%
[25]	CNN + LSTM	CNN used to learn features from the time-series data.	SGCC	9956	Accuracy: 89% F1-score: 58.8%
[26]	LOF + k-means clustering	N/A	SGCC	3500	AUC: 81.5% MAP: 73.35%
[27]	SALM	AlexNet used to extract features from the original time-series data.	SGCC	42372	Accuracy: 90% AUC: 90.6%
[27]	GAN-NetBoost	GoogleNet used to extract features from the original time-series data.	SGCC	42372	Accuracy: 95% AUC: 96%
This work	**Feed forward DNN**	**Manually extracted time-domain and frequency-domain features.**	**SGCC**	**42372**	**Accuracy: 91.8% AUC: 97%**

AUC that is 1% better than the benchmark's best and accuracy that is second-best. The findings demonstrate how well our work stacks up against other recent methodologies.

VI. CONCLUSION

In this study, a DNN-based classification strategy was used to examine how to identify power theft in smart grids utilising time-domain and frequency-domain data. On the same DNN network, separate classification tasks based on time-domain, frequency-domain, and mixed domain features were examined. The model's performance was assessed using commonly used performance metrics including recall, precision, F1-score, accuracy, AUC-ROC, and MCC. We found that classification using features from the frequency domain outperforms classification using features from the time domain, which in turn beats classification using features from both domains.

When put to the test, the classifier managed to attain 87.3% accuracy and 93% AUC-ROC. To reduce features, we utilised PCA. Using 7 of the 20 components, the classifier was successful in attain 92% AUC-ROC and 85.8% accuracy when tested. We then examined how different characteristics in previous studies, and 91.8% accuracy, which is the second highest on the benchmark, when compared to other data-driven algorithms assessed on the same dataset. The approach here makes advantage of consumption data trends. It may be utilised in applications for anomaly detection in any area, in addition to its usage in power distribution networks. We only make a modest dent in correctly detecting energy theft since we only catch theft that happened gradually. Future plans include for expanding our approach to catch power theft in real time. Since a technique to stop power theft in real time in the future. This approach may be further verified against datasets from various places to guarantee its applicability elsewhere because it was assessed based on consumption habits of SGCC consumers.

REFERENCES

[1] S. Foster. (Nov. 2, 2021). *Non-Technical Losses: A $96 Billion Global Opportunity for Electrical Utilities*. [Online]. Available: https://energycentral.com/c/pip/ non-technical-losses-96-billion-global-opportunity-electrical-utilities

[2] Q. Louw and P. Bokoro, ''An alternative technique for the detection and mitigation of electricity theft in South Africa,'' *SAIEE Afr. Res. J.*, vol. 110, no. 4, pp. 209–216, Dec. 2019.

[3] M. Anwar, N. Javaid, A. Khalid, M. Imran, and M. Shoaib, ''Electricity theft detection using pipeline in machine learning,'' in *Proc. Int. Wireless*

Commun. Mobile Comput. (IWCMC), Jun. 2020, pp. 2138–2142.

[4] 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 Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1606–1615, Apr. 2018.

[5] P. Pickering. (Nov. 1, 2021). *E-Meters Offer Multiple Ways to Combat Electricity Theft and Tampering*. [Online]. Available: https://www.electronicdesign.com/technologies/meters

[6] X. Fang, S. Misra, G. Xue, and D. Yang, ''Smart grid—The new and improved power grid: A survey,'' *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 944–980, 4th Quart., 2012.

[7] M. Ismail, M. Shahin, M. F. Shaaban, E. Serpedin, and K. Qaraqe, ''Efficient detection of electricity theft cyber-attacks in AMI networks,'' in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2018, pp. 1–6.

[8] A. Maamar and K. Benahmed, ''Machine learning techniques for energy theft detection in AMI,'' in *Proc. Int. Conf. Softw. Eng. Inf. Manage. (ICSIM)*, 2018, pp. 57–62.

[9] A. Jindal, A. Schaeffer-Filho, A. K. Marnerides, P. Smith, A. Mauthe, and L. Granville, ''Tackling energy theft in smart grids through data-driven analysis,'' in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2020, pp. 410–414.

[10] I. Diahovchenko, M. Kolcun, Z. Čonka, V. Savkiv, and R. Mykhailyshyn, ''Progress and challenges in smart grids: Distributed generation, smart metering, energy storage and smart loads,'' *Iranian J. Sci. Technol., Trans. Electr. Eng.*, vol. 44, no. 4, pp. 1319–1333, Dec. 2020.

[11] M. Jaganmohan. (Mar. 3, 2022). *Global Smart Grid Market Size by Region 2017–2023*. [Online]. Available: https://www.statista.com/statistics/246154/global-smart-grid-market-size-by-region/

[12] Z. Zheng, Y. Yang, X. Niu, H.-N. Dai, and Y. Zhou. (Sep. 30, 2021). *Electricity Theft Detection*, [Online]. Available: https://github.com/henryRDlab/ ElectricityTheftDetection

[13] D. O. Dike, U. A. Obiora, E. C. Nwokorie, and B. C. Dike, ''Minimizing household electricity theft in Nigeria using GSM based prepaid meter,'' *Amer. J. Eng. Res.*, vol. 4, no. 1, pp. 59–69, 2015.

[14] P. Dhokane, M. Sanap, P. Anpat, J. Ghuge, and P. Talole, ''Power theft detection & initimate energy meter information through SMS with auto power cut off,'' *Int. J. Current Res. Embedded Syst. VLSI Technol.*, vol. 2, no. 1, pp. 1–8, 2017.

[15] S. B. Yousaf, M. Jamil, M. Z. U. Rehman, A. Hassan, and S. O. G. Syed, ''Prototype development to detect electric theft using PIC18F452 microcontroller,'' *Indian J. Sci. Technol.*, vol. 9, no. 46, pp. 1–5, Dec. 2016.

[16] K. Dineshkumar, P. Ramanathan, and S. Ramasamy, ''Development of ARM processor based electricity theft control system using GSM net- work,'' in *Proc. Int. Conf. Circuits, Power Comput. Technol. (ICCPCT)*, Mar. 2015, pp. 1–6.

[17] S. Ngamchuen and C. Pirak, ''Smart anti-tampering algorithm design for single phase smart meter applied to AMI systems,'' in *Proc. 10th Int. Conf. Electr. Eng./Electron., Comput., Telecommun. Inf. Technol.*, May 2013, pp. 1–6.

[18] B. Khoo and Y. Cheng, ''Using RFID for anti-theft in a Chinese electrical supply company: A cost-benefit analysis,'' in *Proc. Wireless Telecommun. Symp. (WTS)*, Apr. 2011, pp. 1–6.

[19] J. Astronomo, M. D. Dayrit, C. Edjic, and E. R. T. Regidor, ''Development of electricity theft detector with GSM module and alarm system,'' in *Proc. IEEE 12th Int. Conf. Humanoid, Nanotechnol., Inf. Technol., Commun. Control, Environ., Manage. (HNICEM)*, Dec. 2020, pp. 1–5.

[20] P. Jokar, N. Arianpoo, and V. C. M. Leung, ''Electricity theft detection in AMI using customers' consumption patterns,'' *IEEE Trans. Smart Grid*, vol. 7, no. 1, pp. 216–226, Jan. 2015.

[21] W. Han and Y. Xiao, ''A novel detector to detect colluded non-technical loss frauds in smart grid,'' *Comput. Netw.*, vol. 117, pp. 19–31, Apr. 2017.

[22] S. Sahoo, D. Nikovski, T. Muso, and K. Tsuru, ''Electricity theft detection using smart meter data,'' in *Proc. IEEE Power Energy Soc. Innov. Smart Grid Technol. Conf. (ISGT)*, Feb. 2015, pp. 1–5.

[23] S. K. Singh, R. Bose, and A. Joshi, ''PCA based electricity theft detection in advanced metering infrastructure,'' in *Proc. 7th Int. Conf. Power Syst. (ICPS)*, Dec. 2017, pp. 441–445.

[24] M. Di Martino, F. Decia, J. Molinelli, and A. Fernández, ''Improving elec- tric fraud detection using class imbalance strategies,'' in *Proc. ICPRAM*, 2012, pp. 135–141.

[25] M. N. Hasan, R. N. Toma, A.-A. Nahid, M. M. M. Islam, and J.-M. Kim, ''Electricity theft detection in smart grid systems: A CNN-LSTM based approach,'' *Energies*, vol. 12, no. 17, p. 3310, 2019.

[26] Y. Peng, Y. Yang, Y. Xu, Y. Xue, R. Song, J. Kang, and H. Zhao, ''Electricity theft detection in AMI based on clustering and local outlier factor,'' *IEEE Access*, vol. 9, pp. 107250–107259, 2021.

[27] A. Aldegheishem, M. Anwar, N. Javaid, N. Alrajeh, M. Shafiq, and H. Ahmed, ''Towards sustainable energy efficiency with intelligent electricity theft detection in smart grids emphasising enhanced neural networks,'' *IEEE Access*, vol. 9, pp. 25036–25061, 2021.

[28] K. Phil, *MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence*. Seoul, South Korea: Apress, 2017.

[29] S. Notley and M. Magdon-Ismail, ''Examining the use of neural net- works for feature extraction: A comparative analysis using deep learn- ing, support vector machines, and K-nearest neighbor classifiers,'' 2018, *arXiv:1805.02294*.

[30] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, ''Artificial neural networks-based machine learning for wireless networks: A tutorial,'' 2017, *arXiv:1710.02913*.

[31] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Speech Recognition, Computational Linguistics and Natural Language Processing*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2020.

[32] L. Deng, ''A tutorial survey of architectures, algorithms, and applications for deep learning,'' *APSIPA Trans. Signal Inf. Process.*, vol. 3, no. 1, pp. 1–29, 2014.

[33] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, ''Efficient processing of deep neural networks: A tutorial and survey,'' *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec. 2017.

[34] PARISlab@UCLA. (Jul. 9, 2021). *Training an Artificial Neural Network With MATLAB—Machine Learning for Engineers*. [Online]. Available: https://youtu.be/xOzh6PMk21I

[35] J. Heaton, *Introduction to Neural Networks With Java*, 2nd ed. Chesterfield, U.K.: Heaton Res., 2008.

[36] C. C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*. Cham, Switzerland: Springer, 2018.

[37] J. Zhang, ''Gradient descent based optimization algorithms for deep learning models training,'' 2019, *arXiv:1903.03614*.

[38] B. Ding, H. Qian, and J. Zhou, ''Activation functions and their characteristics in deep neural networks,'' in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2018, pp. 1836–1841.

[39] S. Sharma, S. Sharma, and A. Athaiya, ''Activation functions in neural networks,'' *Towards Data Sci.*, vol. 6, no. 12, pp. 310–316, 2017.

[40] J. Feng and S. Lu, ''Performance analysis of various activation functions in artificial neural networks,'' *J. Phys., Conf. Ser.*, vol. 1237, no. 2, Jun. 2019, Art. no. 022030.

[41] P. Dangeti, *Statistics for Machine Learning*. Birmingham, U.K.: Packt Publishing, 2017.

[42] H. Abdi and L. J. Williams, ''Principal component analysis,'' *Wiley Interdiscipl. Rev., Comput. Statist.*, vol. 2, no. 4, pp. 433–459, 2010.

[43] H. Abdi, ''Singular value decomposition (SVD) and generalized singular value decomposition,'' in *Encyclopedia of Measurement and Statistics*, N. Salkind, Ed. Thousand Oaks, CA, USA: Sage, 2007, pp. 907–912.

[44] M. Billah and S. Waheed, ''Minimum redundancy maximum relevance (mRMR) based feature selection from endoscopic images for automatic gastrointestinal polyp detection,'' *Multimedia Tools Appl.*, vol. 79, nos. 33–34, pp. 23633–23643, Sep. 2020.

[45] C. Ding and H. Peng, ''Minimum redundancy feature selection from microarray gene expression data,'' in *Proc. Comput. Syst. Bioinf., IEEE Bioinf. Conf.*, Aug. 2003, pp. 523–528.

[46] C. Moler. (Mar. 3, 2023). *Splines and Pchips*. [Online]. Available: https://blogs.mathworks.com/cleve/2012/07/16/splines-and-pchips/

[47] G. Dong and H. Liu, *Feature Engineering for Machine Learning and Data Analytics*. Boca Raton, FL, USA: CRC Press, 2018.

[48] B. D. Fulcher and N. S. Jones, ''Highly comparative feature-based time-series classification,'' *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 12, pp. 3026–3037, Apr. 2014.

[49] Codecademy. (Sep. 9, 2021). *Normalization*. [Online]. Available: https://www.codecademy.com/articles/normalization

[50] M. Fitz, *Fundamentals of Communication Systems*. New York, NY, USA: McGraw-Hill, 2007.