IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 November 2023, accepted 28 December 2023, date of publication 2 January 2024,
date of current version 19 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3349352

RESEARCH ARTICLE

Deep Learning-Assisted Energy Prediction Modeling for Energy Harvesting in Wireless Cognitive Radio Devices

OBUMNEME OBIAJULU UMEONWUKA[1], BABATUNDE SEGUN ADEJUMOBI[1], (Member, IEEE), AND THOKOZANI SHONGWE[2], (Senior Member, IEEE)

[1]Department of Electrical and Electronic Engineering Science, University of Johannesburg, Johannesburg 2092, South Africa
[2]Department of Electrical and Electronic Engineering Science, University of Johannesburg, Doornfontein Campus, Johannesburg 2028, South Africa

Corresponding author: Obumneme Obiajulu Umeonwuka (obbtechnix@gmail.com)

This work was supported in part by the University of Johannesburg Library Funds; and in part by the University of Johannesburg Global Excellence and Stature (GES), Fourth Industrial Revolution (4IR) Initiatives.

ABSTRACT Cognitive radio is a technology that allows Secondary Users (SUs) to access vacant spectrum areas allocated to Primary Users (PUs) by dynamically adjusting their settings. However, the spectrum detection subsystem of SUs consumes battery power that could be used for transmission. This work aims to address the energy availability issue for cognitive radio devices by two methods: energy harvesting from the ambient environment and deep learning prediction of future energy levels. We compare three deep learning models: Long-Short Term Memory (LSTM), Convolutional Neural Network (CNN), and Convolutional Long-Short Term Memory (ConvLSTM) with three classic machine learning models: Artificial Neural Networks (ANN), Support Vector Regressor (SVR), and Extreme Gradient Boost (XGBoost). The results show that deep learning models outperform machine learning models across all datasets, with ConvLSTM being the best model with a Normalized Root Mean Squared Error (nRMSE) of 0.0632 and Mean Absolute Error (MAE) of 1.479, which are 8.80% and 9.04% better than the best machine learning model, ANN, with nRMSE of 0.0693 and MAE of 1.626.

INDEX TERMS Cognitive radio networks, deep learning, energy harvesting, machine learning, modeling, wireless communications systems.

I. INTRODUCTION

The radio spectrum includes Radio waves, also known as electromagnetic waves with a frequency range of 3 Hz to 3 THz, which are frequently employed as a transmission medium in modern technologies, including telecommunication. The creation and transmission of radio waves are rigorously governed by regulations that have been standardised by an international organization, the International Telecommunication Union (ITU) [1], to reduce the issue of interference between various users. The total quantity of spectrum is constrained, but technology choices affect how much of it may be used. The region of the electromagnetic spectrum that contains radio waves is known as the radio spectrum [1]. In general, a user can only use radio spectrum after receiving a special license from the appropriate national regulatory organization. Technically speaking, this strategy aids in system design since it is simpler to create a system that functions in a certain band than it is to create a system that can employ several bands dispersed throughout various frequency bands [2].

In Figure 1, the spectrum usage of the radio spectrum is presented. Noticeable from the diagram, there are large portions of sparse use, especially in the lower frequency bands and there are also portions of medium use in the higher frequency bands. Overall, spectral occupancy is less than 6% [3].

The demand for radio spectrum has increased significantly due to recent events. Radio spectrum can be considered as a renewable natural resource that can be reused within

The associate editor coordinating the review of this manuscript and approving it for publication was Yiming Huo.

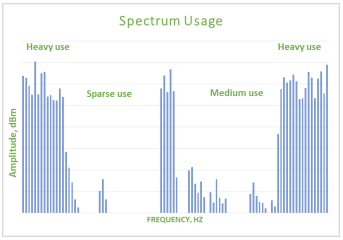

FIGURE 1. Spectrum Usage.

certain constraints under appropriate conditions. However, the spectrum availability is limited and has to support a growing number of connected devices, which creates a lot of pressure on it. The IoT paradigm further exacerbates this spectrum scarcity problem, which makes it urgent to find solutions. One of the proposed solutions in the literature is Cognitive Radio (CR), which can potentially improve spectrum utilization.

Cognitive radio (CR) is a technology that enables dynamic and opportunistic spectrum access by Secondary Users (SUs) who can utilize the frequency bands allocated to Primary Users (PUs) without causing interference. CR relies on software-defined radio (SDR) and smart antennas to adapt its settings to the environment. CR also has a cognitive engine that analyzes the spectrum usage and a policy engine that defines the rules and roles of different users [3], [4], [5], [6], [7]. CR can exploit spectrum holes or white spaces, which are vacant or underutilized parts of the spectrum as seen in Figure 2, by either switching to another band or adjusting its transmission parameters. This is known as Underlay Spectrum Access. CR is a promising solution to improve spectrum efficiency, which is a major challenge for regulatory agencies worldwide due to the limited availability of radio waves and the increasing demand for wireless services [8], [9], [10], [11], [12]. CR has many potential applications in wireless communication systems including mobile and satellite communications, IoT, smart grid, medical applications, military, and security [12], [13], [14], [15], [16]

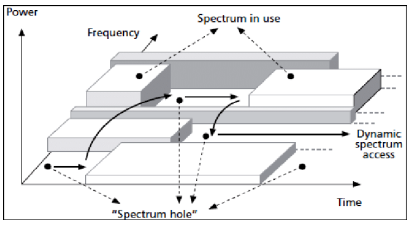

FIGURE 2. Concept of Spectrum Holes.

II. KEY CONTRIBUTIONS OF WORK AND BACKGROUND

The wireless technologies that operate in different frequency bands have different characteristics that affect the temporal correlation of the wireless channel. Temporal correlation is a measure of how the channel conditions change over time, and it depends on factors such as the signal bandwidth, the Doppler spread, the fading model, and the power density. Different wireless technologies may have different protocols, application scenarios, and user behavior that influence these factors [17], [18]. Previous works have used different variations of classical ML and statistical models such as the Autoregressive Integrated Moving Average (ARIMA) and Normalized Least Mean Squares (NLMS) for the prediction, however, these models rely on assumptions about the structure and properties of the time series data, such as stationarity, linearity, autocorrelation, and noise. As such, this makes them limited in terms of broad application and location for highly stochastic data such as harvestable RF energy, which varies widely as the usage pattern on the band of interest, location, and other environmental factors [19], [20]. This work investigates the use of DL models to predict harvested RF energy for CR devices. Among other benefits, DL models place less emphasis on data preprocessing, feature engineering, stationarity, correlation, and noise, making them less susceptible to widely varying data and afford the opportunity for more robust models suitable for highly stochastic RF energy data. To the best of our knowledge, DL models have not been deployed to predict harvested RF energy for CR devices.

III. REVIEW OF SIMILAR WORKS

Since EH-CRNs fall under the category of resource-constrained devices, some of their attributes, such as their computational complexity, speed of execution, energy consumption of computation, memory requirements, and suitability for time-series forecasting, are taken into consideration [21], [22], before a DL algorithm is considered for use in EH CIoT. In this section, we present the current research on DL and ML methods for RF EH wireless networks. It is provided with a synopsis of the findings the researchers came at, along with their methods.

A Fuzzy Neural Network (FNN) was used by the authors of [23] to study the power consumption of system calls made by mobile devices. This FNN was trained to examine process execution behavior based on a series of system call sequence characteristics. They organized the functional components of the system calls into categories, examined each call 100, 000 times, along with the other system parameters, in a database, and then calculated the average energy consumption to create a Power Estimation Daemon (PED) for Linux-based OSs.

According to the findings of their research, there was an overall efficiency of more than 98.8% and a maximum 1.2% discrepancy between the direct power profile of the individual operations and that projected by the PED tool [23]. As the authors in [24] suggested, Dynamic Power Management (DPM), which is a technique to best allocate power to running processes within the device based on demand by such processes other than bulk allocation, can be deployed and used to retain the aggregate power of the system while still harvesting energy for storage and utilization. When there is a low or high energy demand, an action-reward

RL system adjusts the duty cycle allocations for processes while preserving as much of the gathered energy as feasible. As compared to previous adaptive approaches, their findings indicated an increase of over 2.34%.

ML can be deployed as the favoured technique to optimize energy harvested, the amount of energy harvested, harvesting channel, harvesting duration, and energy storage policy, through training and analyzing data using channel state information and other parameters [25]. A coordinated ambient/dedicated RF energy harvesting strategy was presented by the authors in [26]. The system is programmed in this way to collect energy from specialized RF energy sources when they are accessible but to fall back on unintentional (ambient) sources when the dedicated sources are unavailable. They suggested two techniques: the Artificial Neural Network (ANN) and the Linear Forecaster with Near-time Linear Regression-based Enhancer (LFNTLRE) algorithms, for calculating the ideal EH schedule. The LFNTLRE is an algorithm that can adapt to its environment while in progress, but it however requires a limited history of live data to operate making it require more computational memory. ANN is simpler and requires less memory since it can access the history live data from the cloud, instead of locally [26]. Their studies showed that using either the LFNTLRE or the ANN, the chances of getting a source for EH was up to 99.6% and 99.5%, over ideal sensors, respectively. They also showed percentage energy accuracy levels of 100% and 99.0%, respectively. The studies further showed that ANN was better suited for urban areas because the probability of getting unintended sources is lower in rural areas as opposed to urban areas. The LFNTLRE was better suited for the rural areas due to the need for less history live data. However, the performance of ANN closely matched that of the LFNTLRE without the need for live data.

An RF-EH device needs to have a sense of the available energy to better plan and allocate its energy resource to internal processes. Being able to forecast the harvestable energy helps the device to achieve this, and this is the basis of the study by the authors in [27]. Their study focused on harvestable energy prediction for self-powered wearable devices. As they noted, wearables have distinct parametric requirements different from other wireless devices, and their energy management and Quality of Service (QoS) requirements are also different including; execution time, prediction accuracy, and memory, in addition to small and lightweight considerations. They noted that RF energy harvesting has low and sparse power densities and for self-powered devices, being able to predict harvestable energy is key to better managing energy resources and prolonging the lifespan of the device while maintaining a reasonable level of QoS [27].

The authors in [28] employed several ML algorithms to forecast RF energy data using a model approach for effectiveness in wireless communications networks. Using data from the 1805 MHz and 1880MHz bands, they investigated four machine learning (ML) modeling techniques and evaluated their accuracy based on their RF energy models. Linear Regression (LR), Support Vector Machine (SVM), Random Forest Algorithm (RFA), and Decision Tree were the ML algorithms that were investigated. They chose variables such as Feature Length of 1, Number of Observations 120, and Number of Training Splits, which refers to the division between the Training Set and Testing Set. Their results showed that LR performed best for harvesting energy prediction with a mean error of 0.138 under the testing conditions, while RFA performed worse with a mean error of 0.3005, which is unacceptable for real-time prediction purposes. The purposes of harvesting energy prediction are to cater for better planning of more efficient energy harvesters and to reduce overheads in energy consumption of the target devices.

With no CSI information of the intended receiver, the authors in [29] developed a methodology to learn the best power for transmission at the source (for SWIPT) to advance the energy awareness of EH wireless devices. To improve energy efficiency, they created an algorithm based on the Upper Confidence Bound and compared it to a benchmark scheme using CSI knowledge to determine the ideal power level for transmission. Energy availability and management are strained more as a result of CSI estimation's energy use [29]. As a result, their plan, which requires no CSI expertise, performed 52% better than the benchmark plan with CSI. Additionally, the system outperforms the benchmark scheme with CSI knowledge for CSI energy costs higher than $-60dBm$, but underperforms for CSI energy costs below $-60dBm$, which is not practical.

In [30], the authors looked at a DL implementation for sharing resources in battery-operated EH D2D networks. They used the Multi-agent Deterministic Policy Gradient (MADDPG), a POMDP (Partially Observable Markov Decision Process) designed to address the multi-user cooperative task execution problem. In comparison to the baseline algorithms Local and ECLB, their results demonstrated higher performance in terms of lost jobs and overall battery energy penalty. The top total number of dropped jobs for the MADDPG was 110, while for the Local and ECLB, it was 720 and 1300, respectively. The MADDPG outperformed both baseline algorithms, recording 0 (zero) penalty for the total battery energy.

In resource-constrained EH-CRNs, learned resource allocation has been shown to be more effective. Researchers in [31] looked at this issue. To address the issue of the secondary network's primary network's interference threshold, secondary user (SU) QoS, and secondary base station (SBS) restricted power budget, they developed an enhanced DNN-based method. They did this by using transfer learning to take advantage of the initialization of the DNN weights and hasten the convergence of the DNN loss function. Their simulation findings demonstrated improvements over earlier

conventional methods in terms of calculation time, with an improvement of 18.8% over the traditional optimization approaches.

Convex optimization techniques may be used to discover the best power allocation plan for an RF transmitter with many energy harvesters, but the computational complexity required is prohibitive, especially for devices with limited resources. Therefore, fewer computationally intensive strategies are required to solve the optimal power allocation problem; otherwise, the EH device would find itself burning up more CPU cycles and energy merely to obtain the optimal power allocation plan, which would negate the original goal of the plan. The researchers in [32] presented a DL technique suitable for Multiple-Input, Multiple Output (MIMO) applications to examine this research challenge, where the simplified channel vectors are the inputs to the DNN and the network was trained offline using a significant quantity of simulated data. Their findings indicated that their DNN scheme obtained an execution time of 2.47 Sec compared to the convex technique (using CVX), which achieved an execution time of 96.46 sec for the identical transmitter and receiver with three antennas and a $5bps/Hz$ information rate requirement.

Heterogenous Ultra-Dense Networks (HUDN) are a solution to the restricted range of the millimeter waves utilized in these networks for EH devices in the 5G and maybe 6G networks. Due to the expanding number of devices that require even greater data rates, HUDN enables the transmission of millimeter waves. The difficulty in harvesting energy arises from the fact that the EH sources in HUDN are very randomized. The authors of [33] suggested Wolpertinger DDPG (W-DDPG), Deep Deterministic Policy Gradient (DDPG), and Deep Reinforcement Learning (DRL) approaches to address this problem. From the outcomes of their simulations, they found that W-DDPG outperforms the original DDPG and Deep Q-Learning (DQL) algorithms in terms of both energy efficiency and throughput. With a configuration of three macro base stations (MBS), fifty small-cell base stations (SBS), and 100 user equipment (UE), they were able to reach average energy of roughly $11Mbits/Joule$ with W-DDPG throughout 300 episodes, as opposed to the $7Mbits/Joules$ they were able to accomplish with DDPG under the same conditions.

Wireless energy transfer's channel estimation issue has been researched in relation to DL. The authors in [34] suggested a Deep Autoencoder (DAE)-based approach that, in the sense of reducing the Mean Square Error (MSE) of the channel estimate, learns the channel state information (CSI) at the energy transmitter based on the collected energy feedback from the energy receiver. They obtained an MSE of around 0.6 vs an MSE of roughly 1.6 for a Signal-to-Noise Ratio (SNR) of 10db obtained using the traditional method, Gerchberg Saxton algorithm published in [35]. Additionally, given the same SNR of $10dB$, they were able to harvest an average of $6.5Joules$ as opposed to $4.75Joules$ in [35].

IV. ENERGY HARVESTING

The addition of spectrum sensing to wireless devices introduces a new problem, namely the problem of power. The power supply intended for transmission uses provides the necessary power to the sensor circuitry. This limits the amount of electricity needed for CR networks. Additionally, since sensing must take place before transmission can start following spectrum access, sensing shortens the period that is supposed to be reserved for opportunistic transmission. The throughput of the CR device is directly negatively impacted by this.

External energy must be obtained to reduce this. In this field, research has been done on energy harvesting from the surrounding environment. There are several methods for drawing energy from the area around the SU, including solar, wind, piezoelectric, radio frequency energy, etc [36].

A. RADIO FREQUENCY ENERGY HARVESTING

The built environment contains sources of ambient RF energy. In general, these signals are just meant to transmit information and are not meant to be utilized as power sources. However, power may be extracted from these ambient signaling sources and transformed to DC for use by attached devices. The following three things are among the ambient RF energy's sources [37]:

- Broadcast sources such as TV and radio.
- Directed multiuser sources, for example, cellular base station transceivers and WiFi.
- Directed or undirected peer-to-peer wireless links such as Bluetooth, ZigBee, and wireless backhauls.

RF Energy Harvesting (RF-EH) has characteristics, different from harvesting energy from alternate sources, for example, wind, solar, and vibrations, some of these characteristics include:

i Controllable and constant energy transference can be provided by RF sources over a distance for RF energy harvesters.
ii Predictability and relative stability of harvested energy are very common in a fixed RF-EHN over time due to fixed distance due to transmitter footprint.
iii Given that the harvested RF energy amount is proportional to the distance from the RF source, the harvested RF energy of network nodes in the different locations can vary significantly [38].

For energy detection at the SU, the detected signal y received by the SU is seen as a binary hypothesis problem given as [39]:

$$y(t) = \begin{cases} (n(t), H0 \\ g(t)\pi(t) + n(t), H1 \end{cases} \quad (1)$$

where $H0$ and $H1$ represent the respective absence and presence of the PU, respectively, $\pi(t)$ is the PU signal which has a power ps, $n(t)$ is the Gaussian noise and varies as $\sigma^2 n$, from the PU transmitter to the receiver of the SU, the channel

gain is $g(t)$, for sample $t = 1, 2, \ldots, M$. The number of the signal samples is M.

When energy is detected at the antenna input according to equation 1, an ambient RF energy harvester's main goal is to transform the RF energy it receives from the ambient RF sources into DC electricity. These harvesting tools are referred to as Rectenna, which stands for RECtifying anTENNA. As illustrated in Figure 3 [40], [41], a rectenna circuit typically consists of a receiving antenna coupled to an RF bandpass filter, a rectifier, a low-pass filter, and ultimately a load (or energy storage device).

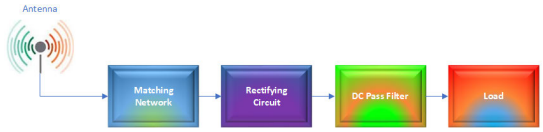

FIGURE 3. Rectenna Block Diagram.

B. POWER CONVERSION EFFICIENCY OF A RECTENNA

The Power Conversion Efficiency (PCE) of a rectenna is a measure of the fraction of the incident power that gets converted to usable energy by the device. This can be given mathematically as [42];

$$\eta = \frac{P_r}{P_i}.100 \tag{2}$$

where η is the efficiency of the rectenna, P_i, is the received signal power in Watts, and P_r is the rectified DC power at the output of the rectenna, in Watts. The useable rectified signal is measured by the PCE as it is given in the equation. Given that some of the received signal is lost in the antenna's circuitry, it tells what percentage of the rectified input signal is useable by the device that it powers. The performance of the rectenna improves with increasing PCE. For a rectenna with a load connected to its output, having a load resistance of R_L, the PCE is given as [43];

$$\eta = \frac{V_r^2}{P_i.R_L}.100 \tag{3}$$

where V_r is the output voltage of the rectifier and R_L is the load resistance in Ohms. This shows that the PCE is inversely proportional to the load resistance, which calls for proper load matching of rectenna to load for optimum power transfer to occur.

The overall ($RF - to - DC$ Power Conversion Efficiency (PCE) is basically determined by the rectenna [44]. Therefore, the amount of electrical energy that can be harvested from incident RF signals depends on the antenna gains of the RF source and that of the receiving device, the power emitted from the RF source, path loss exponent, the distance of the receiving antenna from the RF source antenna, and the $RF - to - DC$ rectification efficiency η_{RF--DC}. The received electrical power is given in [45] as:

$$P_i^{DC} = \eta_{RF--DC} P_i \tag{4}$$

where P_i is the received RF power that can be calculated using the Friis transmission equation. In practical terms, the maximum possible energy transferable wirelessly is limited by various factors such as [46];

- RF Power Restrictions
- RF Distribution features
- Weather related attenuation

The frame structure of an EH-CRN by [47] is shown in Figure 4, where they used a single sensing slot and then an energy harvesting slot per frame. These are followed by a transmit frame which is dependent on the presence, or not, of the PU. When the PU is present, the SU becomes silent.

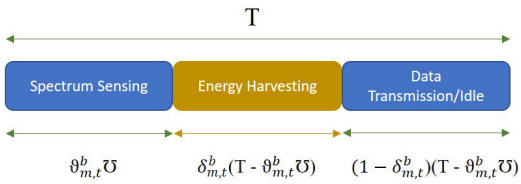

FIGURE 4. Cognitive Radio Energy Harvesting Time Frame Scheme.

For a single band EH-CRN, the sum total of the energy consumed by a SU m, during sensing time slot t, is denoted by $S_{m,t}$ can be given as [47]:

$$S_{m,t} = \vartheta_{m,t} \upsilon P_s \tag{5}$$

where P_s is the sensing circuit power consumption per sample. υ is the sensing time for one sample during sensing time slot t. $\vartheta_{m,t}$ is the number of samples of the sensed spectrum channel. The total harvested energy of SU m during time slot t denoted by $H_{m,t}$ can be written as [47]:

$$H_{m,t} = \delta_{m,t} \eta_{m,t} [(T - \vartheta_{m,t} \upsilon) P_t h_{u,m,t}] \tag{6}$$

where $\eta_{m,t}$ is the RF-to-DC efficiency, T is the frame slot in seconds and ϑ is the sensing time for one sample. $h_{u,m,t}$ is the channel gain during time slot t, between the PU, u, and SU, $\eta_{m,t}$ depends on the frequency of operation and rectification technology.

For energy harvesting to occur over any given band, the average received power from the PU must be greater than a threshold P_{th}, and the sensitivity indicator function, $\delta_{m,t}$ is equal to 1 (when SU m is in HR over band b), under this condition and 0 if the average received power from the PU is less than the threshold P_{th}. This harvesting indicator function $\delta_{m,t}$ is given by [47]:

$P_t h$ is the condition to make sure that the SUs are in HR.

The total harvested energy of SU, m during time slot, t denoted by Hm,t can be written as [47]:

$$H_{m,t} = \Sigma_{b=1}^{B} \delta^b_{m,t} \eta^b_{m,t} [(T - \vartheta^b_{m,t} \upsilon) P^b_t h^b_{u,m,t}] \tag{7}$$

where $\eta^b_{m,t}$ is the RF-to-DC efficiency. T is the frame slot in seconds. υ is the sensing time for one sample.

$\eta^b_{m,t}$ depends on frequency of operation and rectification technology. The stored energy at the end of time slot t at Sm,t,

denoted by $B_{m,t}$, is given by [47]:

$$B_{m,t} = max((B_{m,t-1} + H_{m,t} - S_{m,t}), 0) \quad (8)$$

where $max(x, 0)$ takes a maximum value between x and 0.

C. APPLICATIONS OF RF-EH

One increasingly common use of RF-EHNs is wireless sensor networks. A wireless sensor device's circuit implementation that includes RF energy harvesting can be used as a source of power. The literature has also offered a variety of model structures for sensor devices that use RF energy [38]. The use of RF-powered devices is also possible in medical and healthcare equipment, such as wireless body networks. Thanks to RF-EH, low-power medical devices may obtain work-on-demand power in real-time from specialized RF sources, further allowing small, battery-free circuits. RFID is a common application for RF energy harvesting that has undergone extensive study and has a wide range of uses in inventory management, identification, and tracking. RFID tags, which are like conventional ID cards can have extended long life spans and ranges due to recent advancements in RF energy harvesting technology and low-power circuits [38]. Since low-power mobile devices like wireless keyboards and mice, hearing aids, electronic watches, MP3 players, and other IoT devices typically consume power between microwatts and milliwatts, RF energy harvesting may also be used to provide charging capabilities for these devices. Figure 5 shows a possible implementation of an EH-CR IoT network.

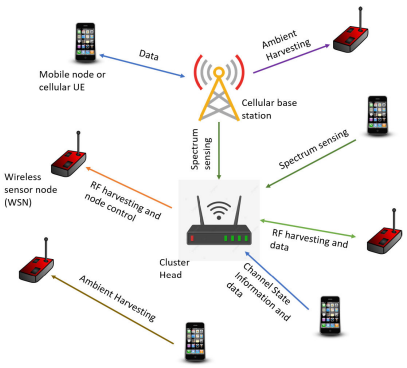

FIGURE 5. A spectrum sharing, energy harvesting and transmitting IoT Network.

V. ENERGY HARVESTING COGNITIVE RADIO NETWORKS WITH MACHINE LEARNING AND DEEP LEARNING TECHNIQUES

The system's optimum performance, device resource allocation, and energy management are all greatly enhanced by the capacity to forecast the usage pattern of the different cognitive device operations. Since RF-EH alone cannot foresee, energy is arbitrarily distributed across the cognitive device's many functions. Machine Learning (ML) offers a chance to increase the spectral efficiency and energy efficiency of the CR device by utilizing energy consumption and channel state prediction.

TABLE 1. A comparison of DL and conventional ML techniques.

Metric	ML Schemes	DL Schemes
Data pre-processing	Highly required	Minimal requirement
Feature engineering	Highly required	Minimally required
Hyper-parameter tuning	Restricted capabilities	Several ways
Data requirement	Require less data	Requires large amounts of data
Hardware requirement	Computationally less (e.g., CPU)	Computationally robust devices (e.g., GPU)
Execution time	Shorter	Longer
Accuracy	Lower	Higher
Complex modelling capability	Weaker	Stronger
Network architecture	Simple	Complex

ML can be used in communications settings when the machine has no prior knowledge of the channel being accessed. A communications system can maximize expected outcomes by optimizing its communications strategy thanks to ML. The ML-enabled node learns by making mistakes in an uncharted environment [48]. In systems with significant co-channel interference, where the Deep Neural Networks (DNN) algorithm is used to optimize interference management by, for example, solving the Weighted Minimum Mean Square Error (WMMSE) problem, ML can also be used to either increase spectral efficiency or optimize the system's overall energy efficiency [48], [49], [50], [51]. Virtualization technology, an aspect of ML has been deployed with significant improvements in energy efficiency and resource utilization which results in up to 50% savings on energy consumed by the overall system [52].

Deep Learning (DL) is increasingly being used to extract even more efficiency from the EH Cognitive devices in terms of managing energy. DL is a branch of AI that uses Artificial Neural Networks (ANN) to compute input functions by propagating the input via weighted layers to the output, simulating the function of neurons in biological systems. By altering the weights of the neural network's layers, the output is learned [53]. DL has grown more appealing as a result of the abundance of data and modern devices' improved processing power [21]. Table 1 shows a comparison of ML and DL schemes based on some general metrics.

A. CLASSIFICATION OF MACHINE LEARNING APPROACHES

ML, which itself is a subfield of Artificial Intelligence (AI), can be loosely classified according to the technique deployed in realizing the desired or expected outcomes. These are Supervised Learning, Unsupervised Learning, Reinforcement Learning, and Hybridized Learning [52], [54]. The performance of ML methods depends on the chosen

features for the particular application. Figure 6 shows the classifications and subcategories of ML.

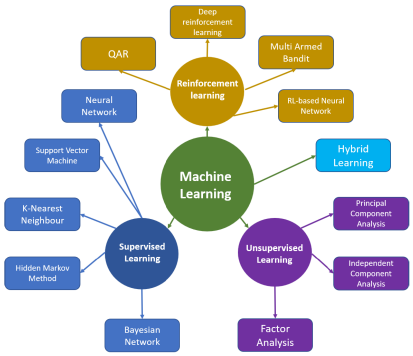

FIGURE 6. Classification and Techniques of Machine Learning for Improved Energy Efficiency of Communication Systems.

VI. DEEP LEARNING MODELS INVESTIGATED IN THIS WORK

In this work, the models investigated were three DL models for RF EH. These are;

1) Long-Short-Term Memory (LSTM)
2) Convolutional Neural Network (CNN)
3) Convolutional Long-Short-Term Memory (ConvLSTM)

A. LONG-SHORT TERM MEMORY (LSTM)

In order to address the issue of back error signals erupting and disappearing in Real Time Recurrent Learning (RTRL) or traditional Back Propagation Through Time (BPTT), both of which are Recurrent Neural Networks, the Long-Short Term Memory (LSTM) was proposed in [55]. Although the forget gate was initially not part of the LSTM network, the authors in [56] proposed it so that the network could reset its state. The three gates regulate the cell's information flow, and the cell retains values across virtually any period. The memory building blocks that make up the LSTM architecture are a group of recurrently linked sub-networks. The memory block's function is to preserve its state over time and regulate data flow using non-linear gating units [57].

B. CONVOLUTIONAL NEURAL NETWORK (CNN)

Convolutional Neural Networks (CNN), also known as ConvNet, are a form of Artificial Neural Networks (ANN) that have a deep feed-forward architecture and exceptional generalizing powers, in contrast to other networks with Fully Connected (FC) layers. It can learn highly abstracted characteristics of things, in particular geographical data, and more accurately identify them [58], [59].

C. CONVOLUTIONAL LONG-SHORT TERM MEMORY (CONVLSTM)

ConvLSTM was initially introduced by Xingjian Shi et al. in [60]. Convolutional structures are present in this model, which is an extension of the Fully Connected LSTM (FC-LSTM) model, in both the input-to-state and state-to-state transitions. By stacking several ConvLSTM layers, they created a network model with an encoding-forecasting structure that is suitable for spatiotemporal sequence forecasting issues. By carrying out the underlying LSTM operation as a convolutional operation, the amount of processing may be significantly reduced [61].

These DL models (LSTM, CNN, and ConvLSTM) were chosen because they are well suited for processing time-correlated, sequences of data and some of the application areas include time series prediction, natural language processing, image and video captioning, text recognition, speech processing and recognition, and so on [57], [62], [63].

D. THE ACTIVATION FUNCTION

The basic objective of any activation function in a neural network-based model is to convert the input to the output, where the input value is produced by aggregating the weighted inputs of all the neurons and biasing the result (if bias exists). The activation function, in other words, decides whether a neuron will be engaged for a specific input by producing the appropriate output. There are several activation functions described in the literature, and the most common ones are Sigmoid, Tanh, and ReLU. ReLU activation function is used in this work because it requires less computation, converges 6 times faster than the other activation functions and has been shown to be more efficient in training deep networks even without pre-training [59], [64], [65].

VII. MACHINE LEARNING MODELS USED IN THIS WORK

To compare the efficiency of the DL models in this work, some popular ML models were also developed. These models and their fundamental concepts are discussed below;

A. SUPPORT VECTOR REGRESSOR

A support vector machine produces a hyperplane or group of hyperplanes in a high- or infinite-dimensional space that may be used for classification, regression, or other tasks. Given that the bigger the margin, the smaller the classifier's overall generalization error is, it seems reasonable that the hyperplane that is farthest from the nearest training data point will achieve good separation for every class. A variant of SVM for regression was first developed in 1996 by Drucker et al. [66]. Support Vector Regression (SVR) is the term for this method and uses actual values for the labels as opposed to classification's binary values [67]. Some of the applications of SVM/SVR include text analysis, image classification, medical sciences, handwriting recognition, and time series forecasting in general.

B. EXTREME GRADIENT BOOST (XGBOOST)

A machine learning technique called gradient tree boosting excels in a wide range of real-world uses. Tree boosting has been demonstrated to deliver cutting-edge outcomes on a variety of standard classification benchmarks. A decision tree ensemble with a high degree of scalability, XGBoost is based

on gradient boosting. Similar to gradient boosting, XGBoost builds an additive extension of the objective function by minimizing a loss function. Because XGBoost only employs decision trees as its fundamental classifiers, a separate loss function is used to control the complexity of the trees [68], [69].

C. ARTIFICIAL NEURAL NETWORK

Artificial neural networks (ANNs) are a class of statistical learning models used in cognitive science and machine learning to estimate or approximation functions that might depend on a large number of inputs and are often unknown. Biological neural networks, notably the brains of animals and their central nervous systems, served as an inspiration for artificial neural networks (ANNs). The nodes (much like biological neurons) are weighted and the data tunneled through a neural network are modified according to the weights of the data [70], [71].

VIII. RESEARCH METHODOLOGY

This section details the methods, working principle, and steps taken in carrying out the research in the development of a DL-assisted energy optimization for energy harvesting CRNs.

A. WORKING PRINCIPLE

In this research, the general process to be implemented in the energy harvesting prediction subsection using LR shall follow the steps outlined in Figure 7 below:

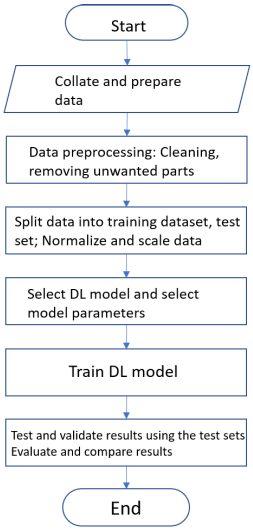

FIGURE 7. Energy Harvesting Prediction Flowchart for Deep Learning Models.

B. DESCRIPTION AND PREPROCESSING OF DATA

The RF energy data in this work was captured for 4 months inside a research laboratory on the University of Warwick campus and obtained from the researchers with permission for academic and personal uses only. The equipment is the Cambridge Radio Frequency Services (CRFS) node. CRFS is a cutting-edge RF designer, with nodes that focus on real-time 24/7 RF spectrum monitoring at a low cost. The Rhode and Schwartz HF907OM antenna was used, a vertically polarized broadband omnidirectional antenna that covers 800 MHz to 26.5 GHz. All of the data was saved in a two-dimensional matrix, with the row representing the time and the column representing the frequency. For example, the band 1805–1880 MHz comprises 448 frequency bins as columns, with each bin having a bandwidth of 0.167 MHz. The data was collected every minute over 131 days (188 917 min) by the authors in [28]. Therefore, the data set of each frequency bin has 188,917-time instants as rows. Eight frequency bands are measured as 880–915 MHz, 925–960 MHz, 1710–1785 MHz, 1805–1880 MHz, 1900–1920 MHz, 1920–1980 MHz, 2110–2170 MHz and 2400–2500 MHz. They represent the U.K. 2G and 3G bands as well as the Wi-Fi band. Comparing these bands, their patterns are similar so we use the 2110–2170 MHz band as an example. Therefore, our predictive models were built for the 2110–2170 MHz band.

Furthermore, for this work, the data subset used in the 2110–2170 MHz band covers data collected every minute over a 30-day period which comprises 43200 samples. This was further subdivided into 1-day, 7-day, and 30-day datasets with sample sizes of 1440, 10,080, and 43,200 timesteps (min), respectively, to give an idea of the predictions for these time durations and also how the data size affects prediction error. These subdivisions of the data are visualized as plots of power level against time steps in Figure 8. A close observation of these plots shows that the data is seasonal with peak and off-peak periods, these can easily be seen as the nighttime when fewer wireless devices are connected, and midday when more devices are connected.

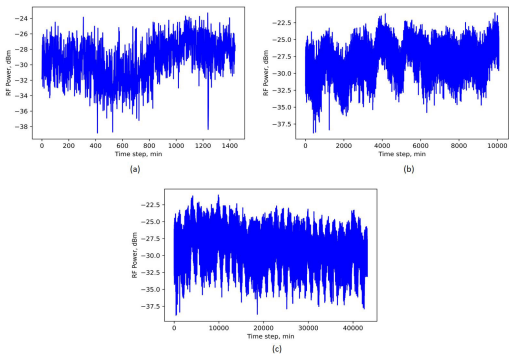

FIGURE 8. Visualization of the data, (a), 1-day data; (b), 7-day data; (c), 30-day data.

Furthermore, a description of the three datasets is presented in Table 2 and shows the various parameters of the data. The data shows that the standard deviation of the 1-day, 7-day and 30-day datasets are 2.72, 2.60 and 2.50, respectively. Also, it further shows that for the 1-day dataset, for example, 25% of the samples lie below $-31.36dBm$, while 50% lie below

$-29.18dBm$ and 75% lie below $-27.20dBm$, with a minimum of $-38.83dBm$ and maximum of $-23.20dBm$.

Due to the observed stochastic nature of the sample, feature scaling was used to minimize the effect of the stochastic data on prediction results while maintaining the core information in each dataset. To do this, Min-Max scaler was used. It was imported from the Sci-kit learn python library and employed on the data for scaling. Figures 9, 10, and 11 show the scatter plots for the three datasets before and after Min Max scaling.

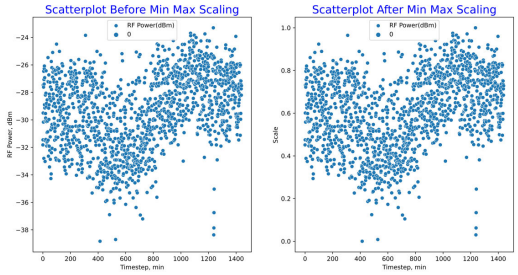

FIGURE 9. 1-day dataset feature scaling; (a) before Min Max scaling; (b) after Min Max scaling.

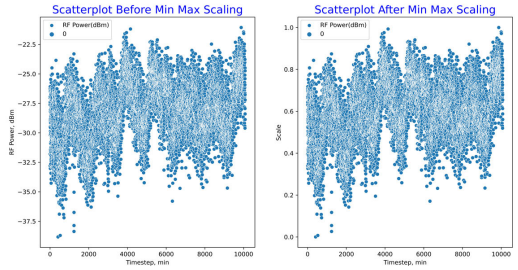

FIGURE 10. 7-day dataset feature scaling; (a) before Min Max scaling; (b) after Min Max scaling.

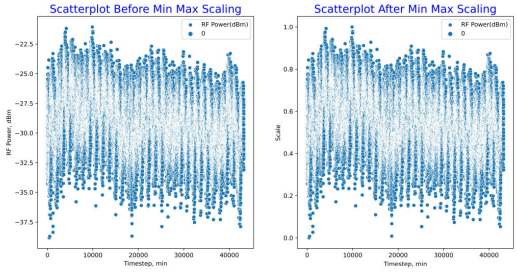

FIGURE 11. 30-day dataset feature scaling; (a) before Min Max scaling; (b) after Min Max scaling.

C. EXPERIMENTAL SET-UP FOR THE DL MODELS

In implementing DL models predictions, the general flowchart of Figure 7 was followed. Using Python programming language running on Jupyter Notebook. The steps are elaborated below and the codes can be found in the Appendix section of this work.

1) Obtaining the dataset and data cleaning: The data was cleaned and unwanted parts were removed. The data was divided into three datasets namely 1-day data, comprising of harvested energy values within 24 hours, every minute. This translates to 1440 minutes or samples. The second dataset was a 7-day data, comprising harvested energy every minute for 7 days, this amounted to 10,080 minutes of samples. The third was a 30-day data, comprising of the energy harvested every minute for 30 days and amounted to 43,200 minutes or samples. The reason for the split was to also investigate the effect of larger data size on prediction error outcomes in DL models. The data were also split in-code into training and test datasets at a ratio of 80:20 and were imported into the Python code using the Pandas library.
2) Training libraries: Anaconda Integrated Development Environment (IDE) was used for this work and elaborated further in subsection F of this section.
3) Feature Scaling/Normalization and reshaping: The scaler used in this work was the MinMaxScaler from the sci-kit learn library. In this study, the training dataset was organized to include 120 timesteps to provide a single output. As a result, the training dataset would include two variables: x-train and y-train, with y-train being the output. The x-train and y-train were converted into number arrays using the NumPy module. Finally, the x-train data's dimension was changed to be in the 3D plane and also lie between 0 and 1 using NumPy.
4) Model Selection: The model was selected by calling the required kernel in Jupyter running the Python code. For this work, the Keras libraries were used for the DL models.
5) Initializing the Model: The model was Sequential with four layers each. Each layer, except the output layer, had 64 units, and returning sequences, and activation was set as ReLU except for the output layer which had Linear activation. Dense was added only on the output layer with just one unit.
6) Compiling the Model: This was achieved by optimization, a stepwise adjustment of the coefficients of the model. To achieve this, the optimizer used was Adam, while the loss metric was set as Mean Squared Error.
7) Fitting the Model to the Training Set: The batch size was set to 32 to fit or train our model utilizing the training dataset. The training dataset set was divided into three separate groups: data from four months, one year, and ten years. At epochs 500 and 1000, each of these groups received unique training with an identical batch size of 32.
8) Testing The Model with Test Dataset: The model testing was done using the test data set of 20% the entire dataset for each category of 1-Day, 7-Day, and 30-Day datasets.
9) Forecasting the Harvested RF Energy: The RF energy harvested was forecasted using the trained model.

TABLE 2. Description of the Datasets; 1-day, 7-day and 30-day Datasets.

Dataset	Count (no. of samples)	Standard deviation	Mean (dBm)	Minimum (dBm)	Maximum (dBm)	25% (dBm)	50% (dBm)	75% (dBm)
1-day	1440	2.72	-29.39	-38.83	-23.20	-31.36	-29.18	-27.20
7-day	10080	2.60	-28.14	-38.83	-21.06	-29.94	-28.03	-26.18
30-day	43200	2.50	-29.04	-38.83	-21.06	-30.81	-28.99	-27.20

TABLE 3. LSTM Model Parameters.

S/N	Parameter	Value
1	Timestep (Look-back)	120
2	Number of batches	32
3	Number of epochs	500, 1000
4	Number of units	50
5	Number of layers	4

TABLE 4. CNN Model Parameters.

S/N	Parameter	Value
1	Timestep (Look-back)	120
2	Number of batches	32
3	Number of epochs	500, 1000
4	Number of units	50
5	Number of layers	4
6	Number of filters	32
7	Number of Nodes	32

10) Visualizing the results: The graphs of all the predictions made using our RNN models were plotted using the Matplotlib package for visualization. The correctness of the data was visualized using independently collected data from epochs 500 and 1000 for 1 day, 7 days, and 30 days. It was found that accuracy rises as the data amount increases.

1) EXPERIMENTAL SETUP OF THE LSTM MODEL

For training and testing of the model, some parameters were defined. Several parameters were selected which are essential for training to occur. These parameters are listed in Table 3.

To use the data, which is time series data, in an LSTM model, it must be reshaped from *samples* to *samples*, *features*, *and timesteps*, to suit the LSTM kernel. This was done using the reshape command in Python.

2) EXPERIMENTAL SETUP OF THE CNN MODEL

For the CNN model, several parameters were selected and these parameters were as given in Table 4;

Also, to use the data which is time series data in a CNN model, it must be reshaped from samples to samples, features, and timesteps, to suit the CNN kernel. This was also done using the reshape command in Python, thus;

$\#reshapeinputtobe[samples, timesteps, features]$
$trainX = np.reshape(trainX, (trainX.shape[0], trainX.shape[1], 1))$
$testX = np.reshape(testX, (testX.shape[0], testX.shape[1], 1))$

TABLE 5. ConvLSTM Model Parameters.

S/N	Parameter	Value
1	Timestep (Look-back)	120
2	Number of batches	32
3	Number of epochs	500, 1000
4	Number of units	50
5	Number of layers	4
6	Number of filters	32
7	Number of Nodes	32

3) EXPERIMENTAL SETUP OF THE CONVLSTM MODEL

For the ConvLSTM model training and testing, some parameters were equally defined. Several parameters were selected which are essential for training to occur. These parameters are listed in Table 5;

To use the data, which is time series data, in a ConvLSTM model, it must be reshaped from samples to samples, features, and timesteps, to suit the ConvLSTM kernel. This was done using the reshape command in Python.

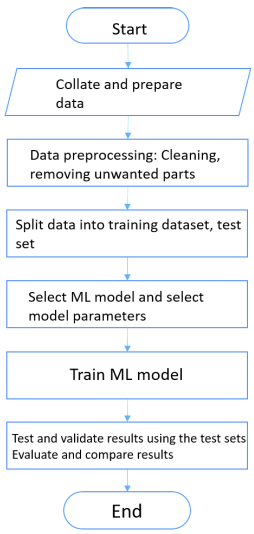

FIGURE 12. Energy Harvesting Prediction Flowchart for Machine Learning Models.

D. EXPERIMENTAL SET-UP FOR THE ML MODELS

Figure 12 shows the Energy harvesting prediction flowchart for machine learning models. The results obtained in the energy prediction subsection shall be compared with the DL algorithms outlined above using the performance metrics.

The steps in Figure 12 in more detail shall follow the process below;

1) Data preprocessing: Collate, sort, and clean parameters data removing anomalies. Remove non-useful data.
2) Import required libraries.
3) Feature Scaling/Normalization and reshaping: The scaler used in this work was the MinMaxScaler from sci-kit learn library. In this study, the training dataset was organized to include 120 timesteps to provide a single output. As a result, the training dataset would include two variables: x-train and y-train, with y-train being the output.
4) Split data into training dataset, test, and validation datasets using a predefined ratio of 80:20, respectively.
5) Instantiate the ML models, calling the required libraries.
6) Select the ML algorithm and implement it to predict harvested energy for a given frequency bin and time.
7) Training data using the ML algorithm.
8) Test and validation of data using the test dataset. Evaluate performance by calculating error rates and applying the performance metrics.
9) End

E. PERFORMANCE METRICS

The performance of all models was evaluated using Root Mean Squared Error (RMSE) and Normalized Root Mean Squared Error (nRMSE). The prediction error which is the Root Mean Square Error (RMSE) for LR for the i-th time slot is given as [72];

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(H_{test}^{i,y} - \hat{H}_{test}^{i,y})^2}{N}} \tag{9}$$

where $\hat{H}_{test}^{i,y}$ represents the predicted target label of the y-th frequency bin evaluated using the trained models and $H_{test}^{i,y}$ represents the actual target label of the harvested power. N is the total number of time slots for the training data, that is total number of samples. The Normalized Root Mean Square Error (NRMSE) is also given as;

$$NRMSE = \frac{RSME}{\frac{\sum_{i=1}^{N}(H_{test}^{i,y})}{N}} \tag{10}$$

Data preparation, training, and testing are the three critical processes that make up the training process. The gathered energy is divided into training and test sets via data preprocessing. We allocate 80% of the captured energy to the training set and 20% to the test set. The training set was used for training, while the test set was used to validate the predictions.

F. SIMULATION ENVIRONMENT

The platform on which the Python coding and simulation was performed was Jupyter Notebook, which is a web-based Python notebook running on Anaconda Navigator 3, which is an Integrated Development Environment (IDE) incorporating various libraries and development platforms for different programming languages, such as Python, R, JavaScript, etc. The required libraries are either already preinstalled on Anaconda Navigator or downloaded and installed. It is worth noting that the libraries used for DL models are provided by Keras. It is an open-source software package and Python interface for artificial neural networks. Its user-friendliness, modularity, and extensibility are its main design goals as it aims to facilitate quick experimentation with deep neural networks.

IX. RESULTS AND ANALYSIS

The findings from the experiments in Section VIII are discussed in this section. In earlier research published in the literature, experiments were conducted to examine the efficacy of ML models for forecasting captured RF energy. However, in this study, trials were conducted to examine how well DL approaches performed on RF energy gathered for wirelessly linked cognitive radio devices. These tests were conducted with safety measures in place to prevent excessive mistake rates and subpar performance. The experiment's findings are shown here, along with evaluations of the data based on the selected performance indicators and conventional ML methods.

A. SIMULATION RESULTS FROM THE LSTM MODEL

As earlier stated in Section VIII, the data was divided into 1-day, 7-day, and 30-day data, and experiments were conducted on them separately. They were then individually subdivided into training and test sets in-code, using train-test-split of Sci kit learn. Below are the results obtained from the three datasets.

1) LSTM 1-DAY DATA RESULTS

The 1-day data consists of samples taken over 24 hours, every minute, which translates to 1440 mins, for 24 hours, i.e. 1440 samples. The model was first trained at 500 epochs, 120 timesteps, and a batch size of 64. The RMSE score obtained on the test data was 1.860dBm which gave an nRMSE of 0.0633. At 1000 epochs, the RMSE score obtained was 1.870, and nRMSE of 0.0636. Figures 13 and 14 show the plot of the actual and predicted values of the model on the test sets.

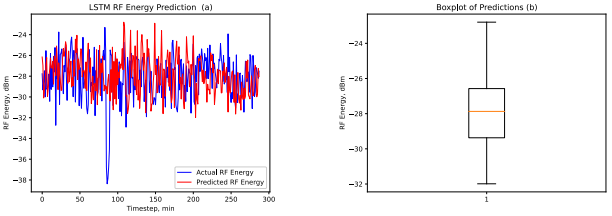

FIGURE 13. 1-day Dataset LSTM model plot of the actual and predicted RF energy harvested values against time at; (a) 500 epochs, RMSE = 1.860dBm, nRMSE = 0.0633; (b) Boxplot distribution of predictions.

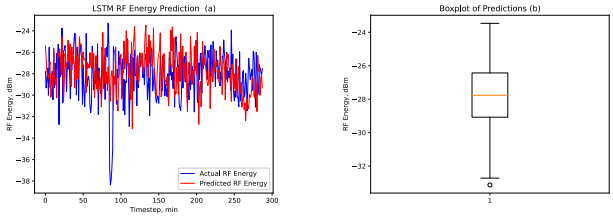

FIGURE 14. 1-day Dataset LSTM model plot of the actual and predicted RF energy harvested values against time at; (a) 1000 epochs, RMSE = 1.870dBm, nRMSE = 0.0636; (b) Boxplot distribution of predictions.

2) LSTM 7-DAY DATA RESULTS

The 7-day data consists of samples taken over 7 days, every minute, which is 10,080 samples in total. Training was performed at 500 and 1000 epochs. The results at 500 epochs showed that RMSE is 1.865dBm and nRMSE of 0.0663. At 1000 epochs, the RMSE was 1.862dBm and nRMSE of 0.0661. The plots of the actual harvested energy and predicted energy against time are shown in Figures 15 and 16 for 500 and 1000 epochs, respectively.

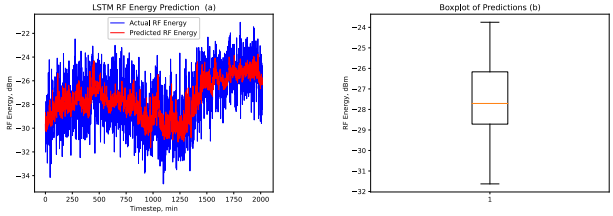

FIGURE 15. 7-day Dataset LSTM model plot of the actual and predicted RF energy harvested values against time at; (a) 500 epochs, RMSE = 1.865dBm, nRMSE = 0.0663; (b) Boxplot of predictions.

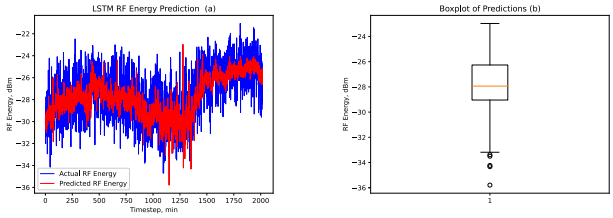

FIGURE 16. 7-day Dataset LSTM model plot of the actual and predicted RF energy harvested values against time at; (a) 1000 epochs, RMSE = 1.862dBm, nRMSE = 0.0661; (b) Boxplot of predictions.

3) LSTM 30-DAY DATA RESULTS

The 30-day data comprises harvested RF energy over 30 days, taken every minute, which translates to 43,200 data samples. Training of the model was performed at 500 epochs and an RMSE of 1.821dBm was recorded and nRMSE of 0.0627. At 1000 epochs, the RMSE was however 1.832dBm and nRMSE of 0.0631. The plots of the actual samples and predicted ones against time are shown in Figures 17 and 18.

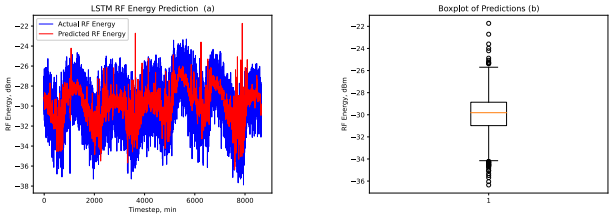

FIGURE 17. 30-Day Dataset LSTM model plot of the actual and predicted RF energy harvested values against time at; (a) 500 epochs, RMSE = 1.821dBm, nRMSE = 0.0627; (b) Boxplot distribution of predictions.

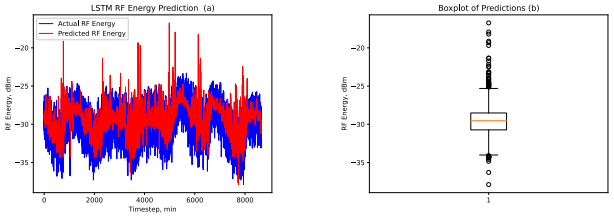

FIGURE 18. 30-Day Dataset LSTM model plot of the actual and predicted RF energy harvested values against time at; (a) 1000 epochs, RMSE = 1.832dBm, nRMSE = 0.0631; (b) Boxplot distribution of predictions.

TABLE 6. Summary of the results for the LSTM model at 500 epochs.

Dataset	RMSE (dBm)	nRMSE	MAE (dBm)
1-day	1.860	0.0633	1.515
7-day	1.865	0.0663	1.513
30-day	1.821	0.0627	1.471

TABLE 7. Summary of the results for the LSTM model at 1000 epochs.

Dataset	RMSE (dBm)	nRMSE	MAE (dBm)
1-day	1.870	0.0636	1.531
7-day	1.862	0.0661	1.504
30-day	1.832	0.0631	1.477

B. SIMULATION RESULTS FROM THE CNN MODEL

Using the same datasets as in the modeling of the LSTM, CNN models were developed and trained by following the steps outlined in SectionVIII.

1) CNN 1-DAY DATA RESULTS

The results of the training performed at 500 epochs were 1.899dBm and an nRMSE of 0.0646. At 1000 epochs, the RMSE was 1.913dBm and nRMSE was 0.0651 The plot of actual and predicted data were carried out and are shown in Figures 19 and 20.

2) CNN 7-DAY DATA RESULTS

Also, the results of training the model with the 7-day data at 500 epochs were 1.863dBm and an nRMSE of 0.0662. At 1000 epochs, the RMSE was 1.866dBm and an nRMSE of 0.0663. The data were plotted and are shown in Figure 21 and 22.

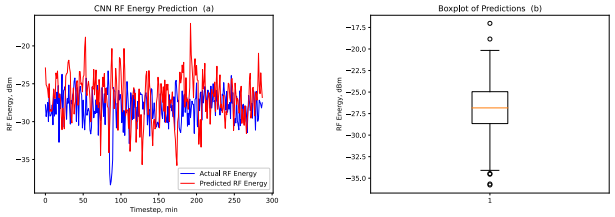

FIGURE 19. 1-day Dataset CNN model plot of the actual and predicted RF energy harvested values of the CNN model against time at; (a) 500 epochs, RMSE = 1.899dBm, nRMSE = 0.0646; (b) Boxplot of predictions.

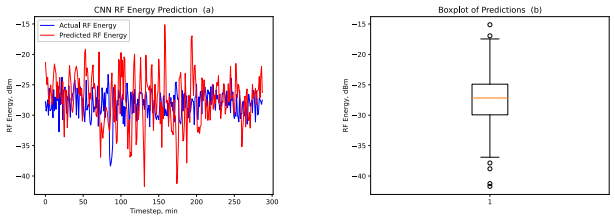

FIGURE 20. 1-day Dataset CNN model plot of the actual and predicted RF energy harvested values of the CNN model against time at; (a) 1000 epochs, RMSE = 1.913dBm, nRMSE = 0.0651; (b) Boxplot of predictions.

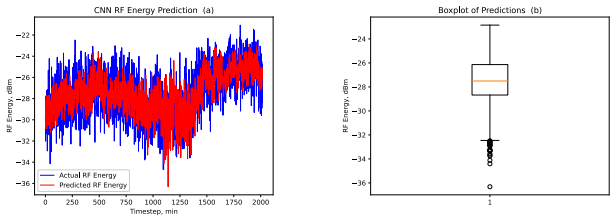

FIGURE 21. 7-day Dataset CNN model plot of the actual and predicted RF energy harvested values of the CNN model against time at; (a) 500 epochs, RMSE = 1.863dBm, nRMSE = 0.0662; (b) Boxplot distribution of predictions.

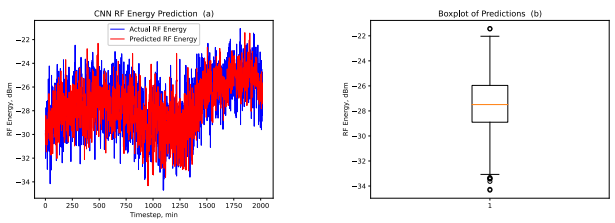

FIGURE 22. 7-day Dataset CNN model plot of the actual and predicted RF energy harvested values of the CNN model against time at; (a) 1000 epochs, RMSE = 1.866dBm, nRMSE = 0.0663; (b) Boxplot distribution of predictions.

3) CNN 30-DAY DATA RESULTS

The simulations were carried out on the 30-day data and it was recorded that at 500 epochs, the RMSE was 1.825dBm and an nRMSE of 0.0629. At 1000 epochs the RMSE was 1.822dBm and an nRMSE of 0.0627. The actual data and prediction results were plotted against time and are shown in Figures 23 and 24.

Tables 8 and 9 show the summary of the results for the CNN model.

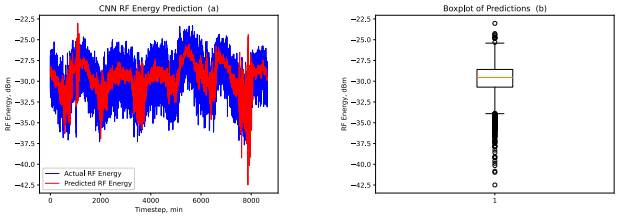

FIGURE 23. 30-day Dataset CNN model plot of the actual and predicted RF energy harvested values of the CNN model against time at; (a) 500 epochs, RMSE = 1.825dBm, nRMSE = 0.0629; (b) Boxplot of predictions.

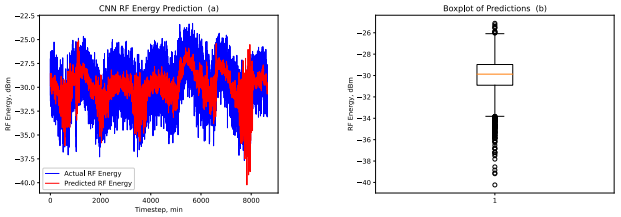

FIGURE 24. 30-day Dataset CNN model plot of the actual and predicted RF energy harvested values of the CNN model against time at; (a) 1000 epochs, RMSE = 1.822dBm, nRMSE = 0.0627; (b) Boxplot of predictions.

TABLE 8. Summary of CNN model results at 500 epochs.

Dataset	RMSE (dBm)	nRMSE	MAE (dBm)
1-day	1.899	0.0646	1.555
7-day	1.863	0.0662	1.502
30-day	1.825	0.0629	1.473

TABLE 9. Summary of the results for the CNN model at 1000 epochs.

Dataset	RMSE (dBm)	nRMSE	MAE (dBm)
1-day	1.913	0.0651	1.542
7-day	1.866	0.0663	1.507
30-day	1.822	0.0627	1.471

C. SIMULATION RESULTS FOR THE CONVLSTM MODEL

Having set up the simulation based on the steps and parameters outlined in Section VIII, the experiments were conducted on the three datasets, and results were obtained. These results are presented below.

1) CONVLSTM 1-DAY DATA RESULTS

For the 1-day dataset, the results obtained at 500 epochs were an RMSE of 1.860dBm and an nRMSE of 0.0633. At 1000 epochs the RMSE obtained was 1.739dBm and nRMSE of 0.0592. The actual data and predicted data were plotted against time and are shown in Figures 25 and 26 for 500 and 1000 epochs, respectively.

4.3.2 ConvLSTM 7-day Data Results The results for the experiment conducted with the 7-day data at 500 epochs were RMSE of 1.850dBm and nRMSE of 0.0657. At 1000 epochs, the RMSE obtained was 1.848dBm and nRMSE of 0.0657. The plots of the actual data and prediction results against

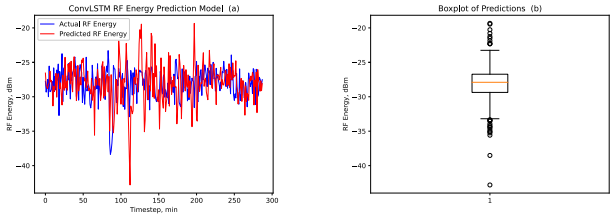

FIGURE 25. 1-day Dataset model Plot of the actual and predicted RF energy harvested values of the ConvLSTM model against time at; (a) 500 epochs, RMSE = 1.860dBm, nRMSE = 0.0633; (b) Boxplot of predictions.

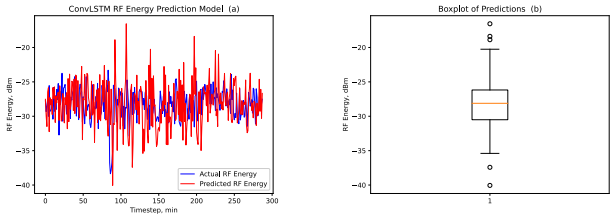

FIGURE 26. 1-day Dataset model Plot of the actual and predicted RF energy harvested values of the ConvLSTM model against time at; (a) 1000 epochs, RMSE = 1.739dBm, nRMSE = 0.0592; (b) Boxplot of predictions.

time were carried out and are shown in Figures 27 and 28 for 500 and 1000 epochs, respectively.

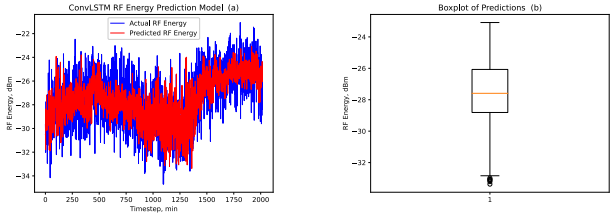

FIGURE 27. 7-day Dataset model plot of the actual and predicted RF energy harvested values of the ConvLSTM model against time at; (a) 500 epochs, RMSE = 1.850dBm, nRMSE = 0.0657; (b) Boxplot distribution of predictions.

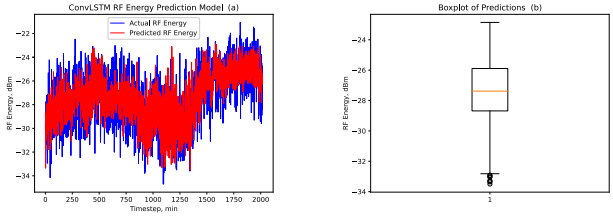

FIGURE 28. 7-day Dataset model plot of the actual and predicted RF energy harvested values of the ConvLSTM model against time at; (a) 1000 epochs, RMSE = 1.848dBm, nRMSE = 0.0657; (b) Boxplot distribution of predictions.

2) CONVLSTM 30-DAY DATA RESULTS

The experimental results for the 30-day data at 500 epochs were an RMSE of 1.821dBm and an nRMSE of 0.0627. At 1000 epochs the RMSE obtained was 1.827dBm and an nRMSE of 0.0629. The plots of the actual data and predicted data were carried out and are shown in Figures 29 and 30 for 500 and 1000 epochs, respectively.

TABLE 10. Summary of ConvLSTM model results at 500 epochs.

Dataset	RMSE (dBm)	nRMSE	MAE (dBm)
1-day	1.860	0.0633	1.518
7-day	1.850	0.0657	1.503
30-day	1.821	0.0627	1.469

TABLE 11. Summary of ConvLSTM model results at 1000 epochs.

Dataset	RMSE (dBm)	nRMSE	MAE (dBm)
1-day	1.739	0.0592	1.469
7-day	1.848	0.0657	1.499
30-day	1.827	0.0629	1.472

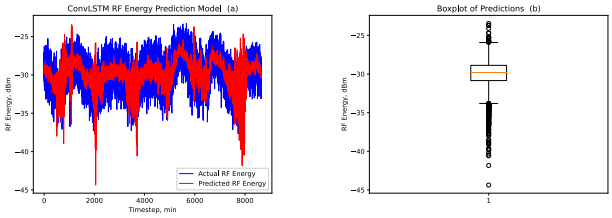

FIGURE 29. 30-day Dataset model plot of the actual and predicted RF energy harvested values of the ConvLSTM model against time at; (a) 500 epochs, RMSE = 1.821dBm, nRMSE = 0.0627; (b) Boxplot distribution of predictions.

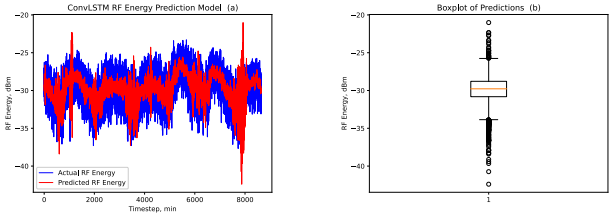

FIGURE 30. 30-day Dataset model plot of the actual and predicted RF energy harvested values of the ConvLSTM model against time at; (a) 1000 epochs, RMSE = 1.827dBm, nRMSE = 0.0629; (b) Boxplot distribution of predictions.

Tables 10 and 11 show the summary of the results for the ConvLSTM model for 500 and 1000 epochs, respectively.

D. SIMULATION RESULTS FOR THE ANN MODEL

In addition to the DL models simulated, three ML models were also developed based on the three datasets as discussed in Section VIII. These models are ANN, SVR, and RF. The model parameters were chosen and the experiments conducted. Below, the results for the ANN model simulations are presented for the datasets.

1) ANN 1-DAY DATA RESULTS

The 1-day data which comprises 1440 data samples was used to train the ANN model and the results obtained for the RMSE were 2.206dBm and an nRMSE of 0.0751. The plots of the

actual data and the predicted data were carried out and are shown in Figure 31.

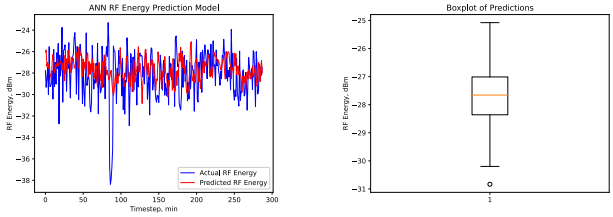

FIGURE 31. 1-day Dataset model Plot of the actual and predicted RF energy harvested values of the ANN model against time at; (a) RMSE = 2.206dBm, nRMSE = 0.0751; (b) Boxplot distribution of predictions.

2) ANN 7-DAY DATA RESULTS

The results obtained when the ANN model was trained with the 7-day dataset, at 1000 epochs was an RMSE of 1.957dBm and an nRMSE of 0.0695. The plots of the actual data and predicted data are shown in Figure 32.

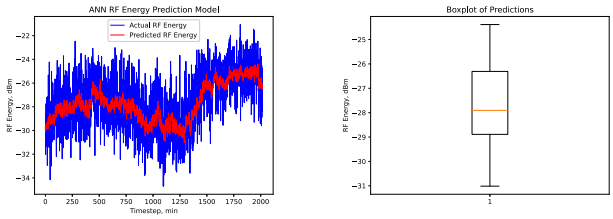

FIGURE 32. 7-day Dataset model plot of the actual and predicted RF energy harvested values of the ANN model against time at; (a) RMSE = 1.957dBm, nRMSE = 0.0695; (b) Boxplot distribution of predictions.

3) ANN 30-DAY DATA RESULTS

The results of training the ANN model using the 30-day dataset were an RMSE of 1.835dBm and an nRMSE of 0.0632 at 1000 epochs. The actual and predicted data were plotted and are shown in Figure 33.

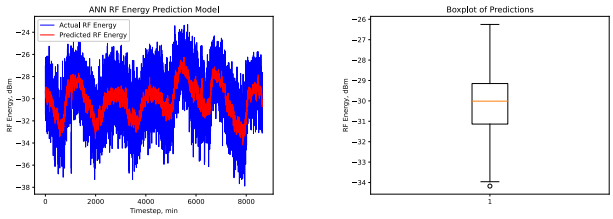

FIGURE 33. 30-day Dataset model plot of the actual and predicted RF energy harvested values of the ANN model against time at; (a) RMSE = 1.835dBm, nRMSE = 0.0632; (b) Boxplot distribution of predictions.

Table 12 shows the summary of the results for the ANN model.

E. SIMULATION RESULTS FOR THE SVR MODEL

The three datasets of 1-day, 7-day, and 30-day, which comprised 1440, 10,080, and 43,200 samples, respectively,

TABLE 12. Summary of ANN model results.

Dataset	RMSE (dBm)	nRMSE	MAE (dBm)
1-day	2.206	0.0751	1.828
7-day	1.957	0.0695	1.567
30-day	1.835	0.0632	1.483

were used to train the SVR model as outlined in Section VIII of this work. The results obtained are presented below.

1) SVR 1-DAY DATA RESULTS

The results obtained using the 1-day dataset for the SVR model were an RMSE of 2.513dBm and an nRMSE of 0.0855. The actual and predicted data were plotted and are shown in Figure 34.

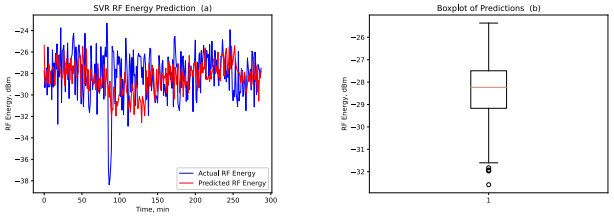

FIGURE 34. 1-day Dataset plot of the actual and predicted RF energy harvested values of the SVR model against time; (a) RMSE = 2.513dBm, nRMSE = 0.0855; (b) boxplot of predictions.

2) SVR 7-DAY DATA RESULTS

The results obtained after training with the 7-day data were an RMSE of 2.197dBm and an nRMSE of 0.0781. The actual and predicted data were plotted and the plot is shown in Figure 35.

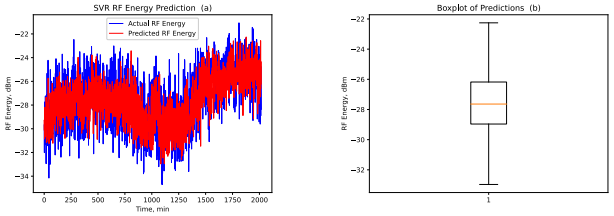

FIGURE 35. 7-day Dataset plot of the actual and predicted RF energy harvested values of the SVR model against time; (a) RMSE = 2.197dBm, nRMSE = 0.0781; (b) Boxplot of predictions.

3) SVR 30-DAY DATA RESULTS

Using the 30-day dataset, the SVR model was trained and an RMSE of 2.091dBm was obtained while an nRMSE of 0.0720 was equally obtained. The actual and predicted results were plotted and are shown in Figure 36.

Table 13 shows the summary of the results for the SVR model.

F. SIMULATION RESULTS OF THE XGBOOST MODEL

Similar to the SVR model training, experiments were conducted on the XGBoost model using the three datasets of

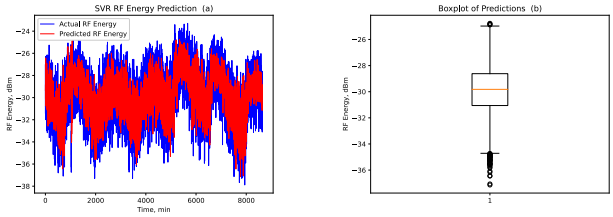

FIGURE 36. 30-day Dataset plot of the actual and predicted RF energy harvested values of the SVR model against time; (a) RMSE = 2.091dBm, nRMSE = 0.0720; (b) Boxplot of predictions.

TABLE 13. Summary of SVR model results.

Dataset	RMSE (dBm)	nRMSE	MAE (dBm)
1-day	2.513	0.0855	1.959
7-day	2.197	0.0781	1.756
30-day	2.091	0.0720	1.666

1-day, 7-day, and 30-day, and results were obtained. These results are presented below.

1) XGBOOST 1-DAY DATA RESULTS

The model training of the XGBoost model with the 1-day dataset yielded results and it was observed that the RMSE obtained was 2.380dBm and an nRMSE of 0.0810. The actual and predicted data were plotted and shown in Figure 37.

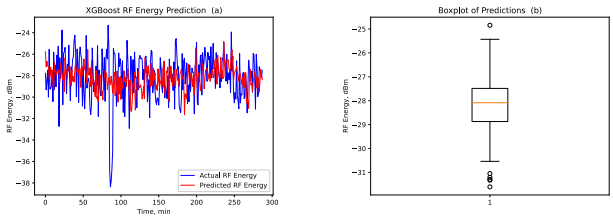

FIGURE 37. 1-day Dataset Plot of the actual and predicted RF energy harvested values of the XGBoost model against time; (a) RMSE = 2.380dBm, nRMSE = 0.0810. (b): Boxplot distribution of predicted values.

2) XGBOOST 7-DAY DATA RESULTS

The performance of the XGBoost model was evaluated using the 7-day dataset and an RMSE of 2.00dBm and an nRMSE of 0.0711 was equally obtained. The plot of the actual and predicted data was plotted against timestep and is shown in Figure 38.

3) XGBOOST 30-DAY DATA RESULTS

The result of simulations for the 30-day data is an RMSE of 1.914dBm and nRMSE of 0.0659. The plot of the actual and predicted data was plotted against timestep and is shown in Figure 39.

Table 14 shows the summary of the results for the XGBoost model.

X. ANALYSIS OF THE RESULTS

In this work, the performance of the models was evaluated using Root Mean Squared Error (RMSE), Normalized Root Mean Squared Error (nRMSE), and Mean Absolute Error (MAE). One of the methods most frequently used to assess the accuracy of forecasts is RMSE, also known as root mean square deviation. It illustrates the Euclidean distance between measured true values and forecasts. When evaluating a model's performance in machine learning, whether during training, cross-validation, or monitoring after deployment, it is very helpful to have a single number. One of the most popular metrics for this is root mean square error. It is an appropriate scoring method that is simple to comprehend and consistent with some of the most widely used statistical presumptions. Absolute error in the context of machine learning refers to the size of the discrepancy between the forecast of an observation and its actual value. The size of errors for the entire group is determined by MAE by averaging the absolute errors for a set of forecasts and observations. MAE is another name for the L1 loss function. nRMSE is obtained by dividing the sum of the RMSE by the difference between the maximum and minimum data points, hence is commonly presented as a percentage.

TABLE 14. Summary of XGBoost model results.

Dataset	RMSE (dBm)	nRMSE	MAE (dBm)
1-day	2.380	0.0810	1.870
7-day	2.00	0.0711	1.602
30-day	1.914	0.0659	1.538

In this work, the DL models investigated performed better than the traditional ML models, across all datasets. More specifically, of all the models investigated, the ConvLSTM performed best with an average nRMSE of 0.0632 and MAE of 1.479 across all datasets, while the LSTM model followed with an nRMSE of 0.0642 and MAE of 1.504, and then the

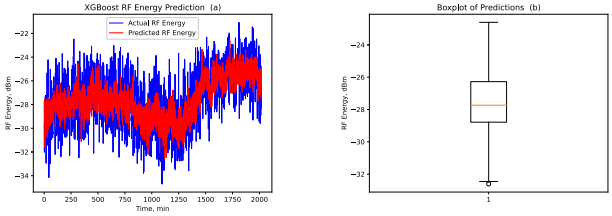

FIGURE 38. 7-day Dataset plot of the actual and predicted RF energy harvested values of the XGBoost model against time; (a) RMSE = 2.00dBm, nRMSE = 0.0711; (b): Boxplot distribution of predicted values.

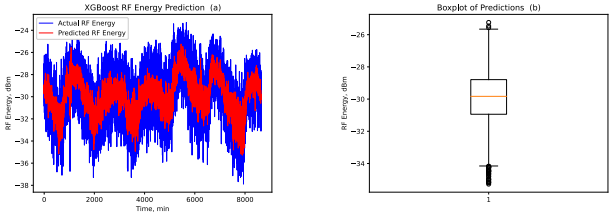

FIGURE 39. 30-day Dataset plot of the actual and predicted RF energy harvested values of the XGBoost model against time, 30-day data; (a) RMSE = 1.914dBm, nRMSE = 0.0659; (b): Boxplot distribution of predicted values.

CNN model with nRMSE of 0.0647 and MAE of 1.507. The least performing model was the SVR with an nRMSE of 0.0785 and MAE of 1.794, this is followed by the XGBoost model with an nRMSE of 0.0723 and MAE of 1.670. The ANN model performed best among the ML models with an nRMSE of 0.0693 and an MAE of 1.626.

Comparing our results to results from previous works, it is worth noting that, to the best of our knowledge, there are few works that deal with RF energy harvesting prediction for CRNs using ML or DL. However, in comparison to the work of researchers in [28] that investigated communications systems in general, show that our models performed better in terms of overall average errors. Specifically, our results show that the ConvLSTM model which achieved an average error of 0.0632 performed better than the best-performing model in [28] which was the LR with an overall average error of 0.139. Furthermore, we showed that our LSTM model obtained an error of 0.0642 compared to the results of the researchers in [28] with an SVM model error of 0.141. Our least-performing model, which was the CNN model performed better than their best-performing model, the LR model, on the overall average, with errors of 0.0647 and 0.139, respectively.

These results show that DL models can perform remarkably better with large datasets requiring least feature engineering and preprocessing, especially when training is performed in the cloud to mitigate the negative effects of high computational complexity.

Figures 40 and 41 show the plots of both the nRMSE and MAE for all models.

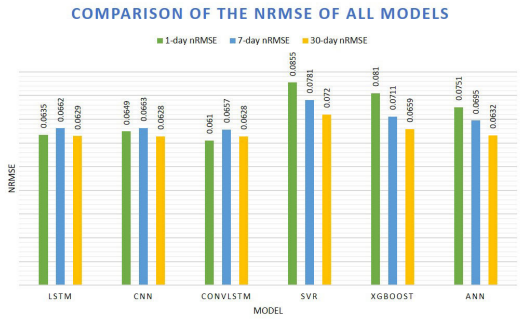

FIGURE 40. Comparison chart of nRMSE values for all models investigated in this work.

These results also show that the DL models performed better than the ML models, with the best-performing DL model, that is, the ConvLSTM having an nRMSE 8.80% superior performance over the best-performing ML model, that is the ANN model nRMSE. The ConvLSTM also had an MAE of 9.04% better than the ANN. Furthermore, the ConvLSTM model had an nRMSE 19.49% better than the SVR, which was the least performing model, and an MAE 17.56% better. Figures 42 and 43 show the plots of average nRMSE and MAE for all models, across all datasets, respectively.

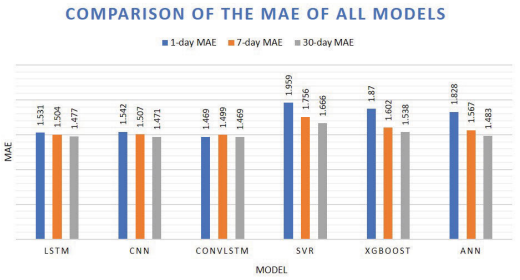

FIGURE 41. Comparison chart of MAE values for all models investigated in this work.

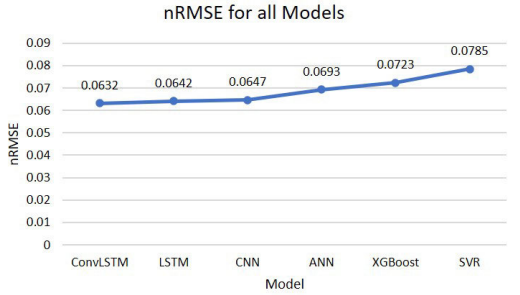

FIGURE 42. Plot of average nRMSE for all Models across all datasets.

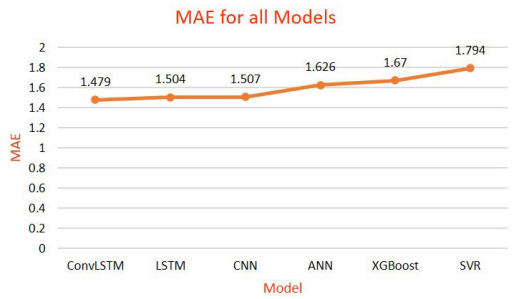

FIGURE 43. Plot of average MAE for all Models across all datasets.

To visualize the performance of the deep learning models with increasing epochs, the loss (which was set as the Mean Squared Error (MSE) in the models) per epoch, up to 1000 epochs was plotted for each dataset set. Figures 44, 45 and 46 show this performance for the 1-day, 7-day and 30-day datasets, respectively.

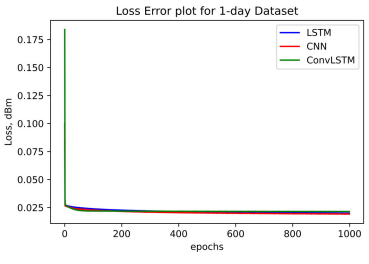

FIGURE 44. Normalized loss per epoch plotted against each epoch for the 1-day dataset, for all three deep learning models.

FIGURE 45. Normalized loss per epoch plotted against each epoch for the 7-day dataset, for all three deep learning models.

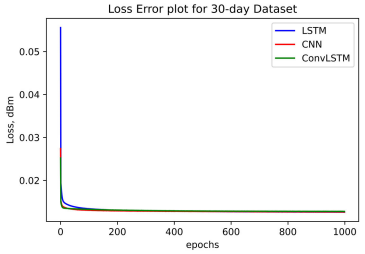

FIGURE 46. Normalized loss per epoch plotted against each epoch for the 30-day dataset, for all three deep learning models.

Furthermore, the MAE per epoch for each dataset was plotted against the total number of epochs up to 1000 and the results are shown in Figures 47, 48 and 49, for the 1-day, 7-day and 30-day datasets, respectively.

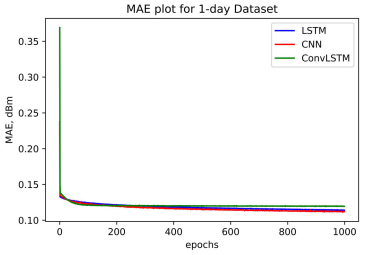

FIGURE 47. Normalized MAE per epoch plotted against each epoch for the 1-day dataset, for all three deep learning models.

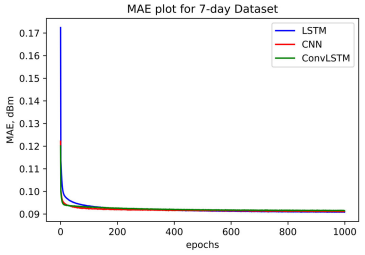

FIGURE 48. Normalized MAE per epoch plotted against each epoch for the 7-day dataset, for all three deep learning models.

It is seen from the plots that beyond about 500 epochs, both the loss and MAE remain almost the same, with only a marginal decrease of up to 1000 epochs. This shows that all three models did not improve much beyond about 500 epochs,

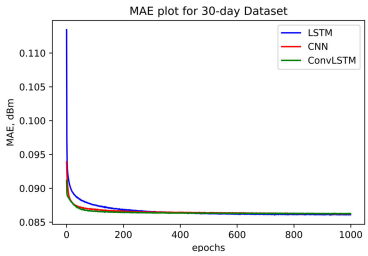

FIGURE 49. Normalized MAE per epoch plotted against each epoch for the 30-day dataset, for all three deep learning models.

and it can be safely concluded that, for this dataset and parameters, 500 epochs are a good choice for training. Beyond that, more epochs only added more complexity and training time, with no significant redeeming benefits on an improved model.

XI. CONCLUSION AND RECOMMENDATIONS

The need for EH-CRNs to be energy-aware and to better allocate their energy resources based on the residual power of its power source, mostly a battery, was discussed. Predicting its available energy from energy harvesting has an indirect impact on the energy savings of the device. In this regard, it was shown through the literature review how ML can help EH-CRNs predict their potential energy resources and as such, better manage and allocate its internal energy-intensive processes, such as spectrum sensing, data transmission and control scheduling, TPC, etc. Already existing work in ML-assisted EH-CRNs was reviewed and the merits and demerits of such schemes were reviewed.

Given their higher efficiencies in prediction tasks, DL schemes were reviewed and existing works showed that they perform better at predicting time series data. However, at the time of writing, no known research covered DL modeling of EH-CRNs, which is a motivation for this work. After simulations were carried out, it was shown that, in general, the DL algorithms reviewed performed better than the ML models in terms of reported error metrics. The ConvLSTM model performed better than all the other models investigated in this work with an nRMSE of 0.0632 and MAE of 1.479. It is followed closely by the LSTM model with an nRMSE of 0.0642 and MAE of 1.504. The least performing DL model among the three was CNN with an nRMSE of 0.0647 and MAE of 1.507. For the ML models, the ANN model performed better than the other two with an nRMSE of 0.0693 and MAE of 1.626, while the XGBoost followed with an nRMSE of 0.0723 and MAE of 1.670. The SVR was the least of all the models with an nRMSE of 0.0785 and MAE of 1.794.

The results show that RF EH-CRNs can benefit more from the superior performance of DL algorithms. The lower the error, the higher the efficiency of the model. This means that devices can better forecast its energy resources and therefore better manage its internal energy processes. These have an indirect impact on the energy savings of the device. It is

worth noting that, even though the computational complexity of DL models is a concern, training cannot always be done on the CR device, rather, it can be done in the cloud, for instance, while inference on the already trained model (which is computationally less expensive by far) can be carried out on the device. As seen for all the models investigated, the more the data, the less prediction error. To take advantage of this, SU CR-IoT devices, which are traditionally data-driven, can very easily collate CSI for each channel, for each spectrum access, and send it to a fusion center fitted with an adaptive model that takes the data and re-trains itself, thereby giving better RF-EH predictions, which the device can use for its internal optimum energy resource management and allocation. These results also make a case for hybrid models for RF EH, as the hybrid ConvLSTM model performed better than the other two stand-alone, traditional DL models.

A. RECOMMENDATIONS FOR FUTURE RESEARCH

In carrying out this research, some challenges were encountered, some were surmounted or surmountable, and others were not. Though the objectives of this work were achieved, some of the challenges encountered can be circumvented in the future, especially as it pertains to DL for EH-CRNs. Therefore, the following recommendations are made;

1) Given the computational complexity and length of time it takes to train DL models, especially with large datasets, training can be done in the cloud on one of the commercial cloud providers available today such as Google Cloud Platform, Microsoft's Azure, Amazon's AWS, Oracle Cloud Infrastructure, etc. These platforms theoretically provide limitless computing power, as opposed to PCs. Training in the cloud will benefit mobile devices as training can be done in the cloud while inference is done at the Edge or Device. By so doing, the computational complexity which is a major drawback of using DL methods in comparison with other methods, would be mitigated.
2) Modeling can be done with the other frequency bands, such as the WiFi band to investigate the performance of the models in such bands.
3) An ensemble DL model such as CNN-LSTM can be used in modeling to further investigate the performance in comparison to the other models and traditional ML models.
4) Real-time RF energy predictions can be researched to investigate the practical performance of the various models.
5) More localized data, such as RF energy harvested in Johannesburg can be used in future work to learn the RF energy footprint of wireless devices in South Africa and to also use that to model RF energy harvesting possibilities for low-power IoT devices in the country.

REFERENCES

[1] *Radio Spectrum*, International Telecommunications Union (ITU), ITU Radio Regulations, Geneva, Switzerland, 2016.
[2] Y. Zeng, Y.-C. Liang, A. T. Hoang, and R. Zhang, "A review on spectrum sensing for cognitive radio: Challenges and solutions," *EURASIP J. Adv. Signal Process.*, vol. 2010, no. 1, pp. 1–15, Dec. 2010.
[3] K. G. Shin, H. Kim, A. W. Min, and A. Kumar, "Cognitive radios for dynamic spectrum access: From concept to reality," *IEEE Wireless Commun.*, vol. 17, no. 6, pp. 64–74, Dec. 2010.
[4] J. Mitola and G. Q. Maguire, "Cognitive radio: Making software radios more personal," *IEEE Pers. Commun.*, vol. 6, no. 4, pp. 13–18, Aug. 1999.
[5] R. A. H. Suliman, K. H. Bilal, and I. Elemam, "Review paper on cognitive radio networks," *J. Electr. Electron. Syst.*, vol. 7, no. 1, pp. 1–3, 2018.
[6] P. S. Jayakrishna, V. Greshma, and T. Sudha, "Energy efficient spectrum sensing techniques for cognitive radio networks: A survey," *Int. J. Comput. Appl.*, vol. 160, no. 4, pp. 20–23, Feb. 2017.
[7] L. Mohjazi, M. Dianati, G. K. Karagiannidis, S. Muhaidat, and M. Al-Qutayri, "RF-powered cognitive radio networks: Technical challenges and limitations," *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 94–100, Apr. 2015.
[8] X. Lu, P. Wang, D. Niyato, and E. Hossain, "Dynamic spectrum access in cognitive radio networks with RF energy harvesting," *IEEE Wireless Commun.*, vol. 21, no. 3, pp. 102–110, Jun. 2014.
[9] L. H. Woo, *Principles of Cognitive Radio*. Cambridge, U.K.: Cambridge Univ. Press, 2013.
[10] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Comput. Netw.*, vol. 50, no. 13, pp. 2127–2159, Sep. 2006.
[11] Md. Z. Alom, T. K. Godder, and M. N. Morshed, "A survey of spectrum sensing techniques in cognitive radio network," in *Proc. Int. Conf. Adv. Electr. Eng. (ICAEE)*, Dec. 2015, pp. 161–164.
[12] J. O. Neel, "Analysis and design of cognitive radio networks and distributed radio resource management algorithms," Ph.D. dissertation, Dept. Elect. Comput. Eng., Virginia Tech, Blackburg, VA, USA, 2006.
[13] S. K. Sharma, T. E. Bogale, L. B. Le, S. Chatzinotas, X. Wang, and B. Ottersten, "Dynamic spectrum sharing in 5G wireless networks with full-duplex technology: Recent advances and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 674–707, 1st Quart., 2018.
[14] M. Lpez-Bentez, *Overview of Recent Applications of Cognitive Radio in Wireless Communication Systems*. Singapore: Springer, 2018, pp. 1–32.
[15] F. Liu, J. Wang, Y. Han, and P. Han, "Cognitive radio networks for smart grid communications," in *Proc. 9th Asian Control Conf. (ASCC)*, Jun. 2013, pp. 1–5.
[16] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, "A roadmap for traffic engineering in SDN-openflow networks," *Comput. Netw.*, vol. 71, pp. 1–30, Oct. 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128614002254
[17] T. S. Rappaport, "Wireless communications–principles and practice, (the book end)," *Microw. J.*, vol. 45, no. 12, pp. 128–129, 2002.
[18] A. F. Molisch, *Wireless Communications*, vol. 34, Hoboken, NJ, USA: Wiley 2012.
[19] M. Bkassiny, Y. Li, and S. K. Jayaweera, "A survey on machine-learning techniques in cognitive radios," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1136–1159, 3rd Quart., 2013.
[20] I. H. Sarker, "Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions," *Social Netw. Comput. Sci.*, vol. 2, no. 6, p. 420, Nov. 2021.
[21] F. Jiang, W. Yi, S. Li, B. Zhu, and W. Yu, "Joint optimization of spectrum sensing and energy harvesting for cognitive radio network," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. Appl. Int. Conf. Ubiquitous Comput. Commun. (ISPA/IUCC)*, Dec. 2017, pp. 423–427.
[22] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2923–2960, 4th Quart., 2018.
[23] D.-R. Chen, Y.-S. Chen, L.-C. Chen, M.-Y. Hsu, and K.-F. Chiang, "A machine learning method for power prediction on the mobile devices," *J. Med. Syst.*, vol. 39, no. 10, pp. 1–11, Oct. 2015.
[24] R. C. Hsu, C.-T. Liu, and W.-M. Lee, "Reinforcement learning-based dynamic power management for energy harvesting wireless sensor network," in *Proc. IEA/AIE 2nd Int. Conf. Ind.*, Cham, Switzerland: Springer, Jun. 2009, pp. 399–408.
[25] H. Yu and M. J. Neely, "Learning-aided optimization for energy-harvesting devices with outdated state information," *IEEE/ACM Trans. Netw.*, vol. 27, no. 4, pp. 1501–1514, Aug. 2019.

[26] J. C. Kwan, J. M. Chaulk, and A. O. Fapojuwo, "A coordinated ambient/dedicated radio frequency energy harvesting scheme using machine learning," *IEEE Sensors J.*, vol. 20, no. 22, pp. 13808–13823, Nov. 2020.

[27] M. A. Wahba, A. S. Ashour, and R. Ghannam, "Prediction of harvestable energy for self-powered wearable healthcare devices: Filling a gap," *IEEE Access*, vol. 8, pp. 170336–170354, 2020.

[28] Y. Ye, F. Azmat, I. Adenopo, Y. Chen, and R. Shi, "RF energy modelling using machine learning for energy harvesting communications systems," *Int. J. Commun. Syst.*, vol. 34, no. 3, Feb. 2021, Art. no. e4688.

[29] D. Ghosh, M. K. Hanawal, and N. Zlatanov, "Learning to optimize energy efficiency in energy harvesting wireless sensor networks," *IEEE Wireless Commun. Lett.*, vol. 10, no. 6, pp. 1153–1157, Jun. 2021.

[30] B. Huang, D. Yu, L. Pan, Z. Li, H. Hu, and V. Chang, "Deep learning based multiple energy harvesting users cooperative task execution," in *Proc. IEEE 22nd Int. Conf. High Perform. Comput. Commun.; IEEE 18th Int. Conf. Smart City; IEEE 6th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Dec. 2020, pp. 764–771.

[31] H. Hu, C. Yang, D. Wu, and R. Q. Hu, "DNN-based resource allocation for cooperative Cr networks with energy harvesting," in *Proc. IEEE 93rd Veh. Technol. Conf. (VTC-Spring)*, Apr. 2021, pp. 1–6.

[32] Y. Xing, Y. Qian, and L. Dong, "Deep learning for optimized wireless transmission to multiple RF energy harvesters," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Aug. 2018, pp. 1–5.

[33] J. Ye and H. Gharavi, "Deep reinforcement learning-assisted energy harvesting wireless networks," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 2, pp. 990–1002, Jun. 2021.

[34] J.-M. Kang, C.-J. Chun, and I.-M. Kim, "Deep-learning-based channel estimation for wireless energy transfer," *IEEE Commun. Lett.*, vol. 22, no. 11, pp. 2310–2313, Nov. 2018.

[35] R. W. Gerchberg, "A practical algorithm for the determination of phase from image and diffraction plane pictures," *Optik*, vol. 35, pp. 237–246, Sep. 1972.

[36] U. Muncuk, K. Alemdar, J. D. Sarode, and K. R. Chowdhury, "Multi-band ambient RF energy harvesting circuit design for enabling battery-less sensors and IoT," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2700–2714, Mar. 2018.

[37] A. N. Parks, "Employing ambient RF energy in connected devices," Ph.D. dissertation, Dept. Electron. Eng., Univ. Washington, Seattle, WA, USA, 2017.

[38] X. Lu, P. Wang, D. Niyato, D. I. Kim, and Z. Han, "Wireless networks with RF energy harvesting: A contemporary survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 757–789, 2nd Quart., 2015.

[39] H. Liu, J. Chen, G. Ding, T. A. Tsiftsis, and C. Rowell, "Antenna beamforming for energy harvesting in cognitive radio networks," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Mar. 2016, pp. 1–4.

[40] V. Kuhn, C. Lahuec, F. Seguin, and C. Person, "A multi-band stacked RF energy harvester with RF-to-DC efficiency up to 84%," *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 5, pp. 1768–1778, May 2015.

[41] O. O. Umeonwuka, B. S. Adejumobi, and T. Shongwe, "Rectenna designs for energy harvesting IoT devices: Overview of current trends and future directions," in *Proc. Int. Conf. Electr., Comput. Energy Technol. (ICECET)*, Dec. 2021, pp. 1–6.

[42] A. Okba, A. Takacs, and H. Aubert, "Compact flat dipole rectenna for IoT applications," *Prog. Electromagn. Res. C*, vol. 87, pp. 39–49, 2018.

[43] M. Aboualalaa and H. Elsadek, "Rectenna systems for RF energy harvesting and wireless power transfer," in *Recent Wireless Power Transfer Technologies*. London, U.K.: IntechOpen, 2020, pp. 1–24.

[44] S. K. Divakaran, D. D. Krishna, and Nasimuddin, "RF energy harvesting systems: An overview and design issues," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 29, no. 1, Jan. 2019, Art. no. e21633.

[45] H. Sharma and P. Agrawal, "A survey of energy harvesting technologies for wireless sensor networks," Ajay Kumar Garg Eng. College, 2019, pp. 46–51. [Online]. Available: https://akgec.ac.in/wp-content/uploads/2019/06/AKGEC_Intel_Journal_8_no_1_9.pdf

[46] Y. Luo, L. Pu, G. Wang, and Y. Zhao, "RF energy harvesting wireless communications: RF environment, device hardware and practical issues," *Sensors*, vol. 19, no. 13, p. 3010, Jul. 2019.

[47] A. Alsharoa, N. M. Neihart, S. W. Kim, and A. E. Kamal, "Multi-band RF energy and spectrum harvesting in cognitive radio networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[48] A. Masadeh, L. Ruan, and S. W. Kim, "Enhancing the performance of energy harvesting wireless communications using optimization and machine learning," Ph.D. dissertation, Dept. Elect. Comput. Eng., Iowa State Univ., Ames, IA, USA, 2019.

[49] K. W. Choi, S. I. Hwang, A. A. Aziz, H. H. Jang, J. S. Kim, D. S. Kang, and D. I. Kim, "Simultaneous wireless information and power transfer (SWIPT) for Internet of Things: Novel receiver design and experimental validation," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2996–3012, Apr. 2020.

[50] O. O. Umeonwuka, B. S. Adejumobi, and T. Shongwe, "Deep learning algorithms for RF energy harvesting cognitive IoT devices: Applications, challenges and opportunities," in *Proc. Int. Conf. Electr., Comput. Energy Technol. (ICECET)*, Jul. 2022, pp. 1–6.

[51] A. H. Sodhro, M. S. Obaidat, S. Pirbhulal, G. H. Sodhro, N. Zahid, and A. Rawat, "A novel energy optimization approach for artificial intelligence-enabled massive Internet of Things," in *Proc. Int. Symp. Perform. Eval. Comput. Telecommun. Syst. (SPECTS)*, Jul. 2019, pp. 1–6.

[52] A. Mughees, M. Tahir, M. A. Sheikh, and A. Ahad, "Towards energy efficient 5G networks using machine learning: Taxonomy, research challenges, and future research directions," *IEEE Access*, vol. 8, pp. 187498–187522, 2020.

[53] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for Internet of Things (IoT) security," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1646–1685, 3rd Quart., 2020.

[54] R. P. Paiva, "Machine learning: Applications, process and techniques," Univ. Coimbra, Portugal, Tech. Rep., 2013.

[55] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[56] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000.

[57] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artif. Intell. Rev.*, vol. 53, pp. 5929–5955, Dec. 2020.

[58] A. Ajit, K. Acharya, and A. Samanta, "A review of convolutional neural networks," in *Proc. Int. Conf. Emerg. Trends Inf. Technol. Eng. (IC-ETITE)*, Feb. 2020, pp. 1–5.

[59] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, "Fundamental concepts of convolutional neural network," *Recent Trends and Advances in Artificial Intelligence and Internet of Things*. Switzerland: Springer, 2020, pp. 519–567.

[60] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, Jun. 2015, pp. 1–11.

[61] K.-S. Kim, J.-B. Lee, M.-I. Roh, K.-M. Han, and G.-H. Lee, "Prediction of ocean weather based on denoising AutoEncoder and convolutional LSTM," *J. Mar. Sci. Eng.*, vol. 8, no. 10, p. 805, Oct. 2020.

[62] T. J. Saleem and M. A. Chishti, "Deep learning for the Internet of Things: Potential benefits and use-cases," *Digit. Commun. Netw.*, vol. 7, no. 4, pp. 526–542, Nov. 2021.

[63] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. Interspeech*, 2014, pp. 338–342.

[64] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018.

[65] J. Wu, "Introduction to convolutional neural networks," Nat. Key Lab Novel Softw. Technol., Nanjing Univ., Nanjing, China, 2017, p. 495, vol. 5, no. 23. [Online]. Available: https://cs.nju.edu.cn/wujx/paper/CNN.pdf

[66] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 1996, pp. 1–7.

[67] A. Smola and S. Vishwanathan, *Introduction to Machine Learning*. vol. 32, no. 34. Cambridge, U.K.: Cambridge Univ., 2008.

[68] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[69] S. Malik, R. Harode, and A. Kunwar, *Xgboost: A Deep Dive Into Boosting*. Burnaby, BC, Canada: Simon Fraser Univ., 2020, pp. 1–21.

[70] S. Shanmuganathan, *Artificial Neural Network Modelling: An Introduction*. Cham, Switzerland: Springer, 2016.

[71] H. Allende, C. Moraga, and R. Salas, "Artificial neural networks in time series forecasting: A comparative analysis," *Kybernetika*, vol. 38, no. 6, pp. 685–707, 2002.
[72] F. Azmat, Y. Chen, and N. Stocks, "Predictive modelling of RF energy for wireless powered communications," *IEEE Commun. Lett.*, vol. 20, no. 1, pp. 173–176, Jan. 2016.

OBUMNEME OBIAJULU UMEONWUKA received the B.Eng. degree in electrical and electronic engineering from Anambra State University, Anambra, Nigeria, in 2008. He is currently pursuing the M.Eng. degree with the Electrical and Electronic Engineering Program, University of Johannesburg, South Africa.

His research interests include artificial intelligence, machine learning for communications systems, the IoT, energy harvesting, and cognitive radio networks. He received the prestigious University of Johannesburg Global Excellence Stature and Fourth Industrial Revolution (GES 4.0) Award, in 2021, for his M.Eng. research in the department. He was also a recipient of the Postgraduate Merit Award, which is a merit-based scholarship to carry out research geared toward the M.Eng. degree.

BABATUNDE SEGUN ADEJUMOBI (Member, IEEE) received the B.Sc. and M.Sc. degrees in electronic and computer engineering from Lagos State University, Lagos, Nigeria, in 2007 and 2012, respectively, and the Ph.D. degree in electronic engineering (wireless communication) from the Department of Electrical, Electronic, and Computer Engineering, University of KwaZuluNatal, Durban, South Africa.

His current research interests include spatial modulation, space-time block-coded modulation, orthogonal frequency division multiplexing, optical communication, wireless communications, energy harvesting, AI techniques, and image processing.

THOKOZANI SHONGWE (Senior Member, IEEE) received the B.Eng. degree in electronic engineering from the University of Swaziland, Swaziland, in 2004, the M.Eng. degree in telecommunications engineering from the University of the Witwatersrand, South Africa, in 2006, and the D.Eng. degree from the University of Johannesburg, South Africa, in 2014.

He is currently an Associate Professor in telecommunications and the Vice Dean of Research, Postgraduate Studies, and Internationalization with the Faculty of Engineering and the Built Environment, University of Johannesburg, and the Co-Founder of a research group with the University of Johannesburg called artificial intelligence for electrical engineering applications (AI for EE applications). This research group is currently composed of a five staff member; a two postdoctoral researcher working in the fields of communications and machine learning; and five doctoral students and ten master's degree students working in the fields of power line communications, visible light communications, application of ML in PLC, power systems, agriculture, and object tracking. His research interests include digital communications, error correcting coding, power-line communications, cognitive radio, smart grids, visible light communications, machine learning, and artificial intelligence.

Prof. Shongwe was a recipient of the 2014 University of Johannesburg Global Excellence Stature (GES) Award, which was awarded to him to carry out his postdoctoral research with the University of Johannesburg. In 2016, he was a recipient of the TWAS-DFG Cooperation Visits Programme funding to do research in Germany. Other awards that, he has received in the past include the post-graduate merit award scholarship to pursue the master's degree with the University of the Witwatersrand, in 2005, which is awarded on a merit basis. In 2012, he (and his coauthors) received an award of the best student paper at the IEEE ISPLC 2012 (Power Line Communications Conference) in Beijing, China.