Engineering Applications of Artificial Intelligence 133 (2024) 108232

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

A complete artificial intelligence pipeline for radio frequency energy prediction in cellular bands for energy harvesting systems

Shaimaa H. Mohammed [a,*], Ashraf S. Mohra [a], Ashraf Y. Hassan [a], Ahmed F. Elnokrashy [a,b,**]

[a] Electrical Engineering Department, Benha Faculty of Engineering, Benha University, Benha, 13511, Egypt
[b] Computer Science Department, Faculty of Information Technology and Computer Science, Nile University, Giza, 12677, Egypt

ARTICLE INFO

Keywords:
Radio frequency
Energy harvesting
Energy prediction
Machine learning
Time series model
Loss function

ABSTRACT

Radio Frequency (RF) energy harvesting has been used to power wireless and low-powered devices. However, RF energy harvesting has limitations in terms of the amount of power that can be collected based on signal availability. Hence, energy prediction is essential to improve energy harvesting circuits' performance. Previous research has mainly focused on improving power harvesting policies or theoretically estimating the harvested energy. Very few works have considered the prediction of the RF signal as time series data using real RF measurements. Moreover, challenges such as the power consumed by the circuit's harvesting decisions and the impact of outliers on the model performance haven't been addressed yet. This paper presents a complete pipeline for developing the best predictive model for RF energy in cellular frequency bands. Real-time measurements are taken in different frequency bands using software-defined radio technology. We use four artificial intelligence techniques to model the RF energy signal. Additionally, we propose an optimized model with an enhanced loss function, which makes the model more resilient to anomalies, saving computational power and time consumed in cleaning the data. The four algorithms are investigated, and their prediction accuracies are compared. The average power of a period of 5 min is accurately forecasted. Numerical results in the 1960 MHz band show that long short-term memory has the best performance, followed by the DeepAR algorithm with prediction accuracies of 95.76% and 95.02%, respectively. Moreover, the proposed optimized model showed a 32.2% lower prediction error than the traditional models.

1. Introduction

Energy harvesting technology has replaced traditional batteries since it is a clean and environmentally friendly source. Energy can be scavenged from several sources, which include solar energy, vibrational energy, wind energy, thermal energy, and radio frequency energy (Sil et al., 2017). In particular, radio frequency energy is a reasonable choice as it can be harvested from TV signals, radio waves, Wi-Fi signals, and satellite stations (Mekid et al., 2017). RF energy harvesting technology can be described as the procedure of collecting RF signals from ambient or dedicated sources and converting them into electrical power. Since it is available, small in size, and implantable, radio frequency energy harvesting (RFEH) has been used in cognitive radio networks, wireless sensor network (WSN) technologies (Varghese et al., 2016), (Mouapi and Hakem, 2018), biomedical wearable devices (Hesham et al., 2021), and internet of things (IoT) applications (Tran et al., 2017), (Ozger et al., 2018).

RF energy harvesting has been used in various industries and applications. For example, An RF energy harvester was developed to harvest from the Wi-Fi signals at the 2.45 GHz band and could generate direct current (DC) power even with low power inputs of −40 dB m (Olgun et al., 2012). The generated power was sufficient to drive a temperature and humidity meter with Liquid Crystal Display (LCD) display. Another energy harvester in (Vyas et al., 2013) designed to harvest from a wireless TV signal could power a 16-bit microcontroller. Researchers in (Syed et al., 2017) have also designed an energy harvesting chip that could harvest at an input power of −12 dB m, which was used to run a microcontroller + radio System on Chip (SoC). The study in (Khan et al., 2020) proposes a reconfigurable 2.45 GHz RF-DC power converter to harvest RF energy. The circuit could switch between a low-power path and a high-power path based on the RF input power. The circuit could achieve more than 20% efficiency. In industry, the

* Corresponding author.
** Corresponding author. Electrical Engineering Department, Benha Faculty of Engineering, Benha University, Benha, 13511, Egypt.
E-mail addresses: Shimaa.hassan@bhit.bu.edu.eg (S.H. Mohammed), ahmed.elnokrashy@bhit.bu.edu.eg (A.F. Elnokrashy).

https://doi.org/10.1016/j.engappai.2024.108232
Received 24 December 2022; Received in revised form 11 January 2024; Accepted 5 March 2024
Available online 12 March 2024

Powercast P2110 harvester can achieve output power of up to 5.25 V (Powercast). Another RF energy harvester has been used to power a sensor module in a food monitoring quality system. (Do et al., 2021), (Lam et al., 2020). The designed harvester operates at 915 MHz frequency and could output 3.3 V with a low input power of −8 dB. In IoT applications, an RF energy harvester at 1.8 GHz was designed to power sensor nodes in a museum monitoring system (Eltresy et al., 2019). Researchers in (Xu et al., 2019) presented a design of an RF energy harvester at 2.45 GHz to supply an IoT smart sensor system. The proposed design could achieve 48.3% efficiency at −3 dB m input power.

On the contrary, RF energy is time, location, and spectrum-dependent. The amount of RF energy on working days differs from weekends for a specific location. Also, some frequency bands have more RF energy than others. Places with more people using cell phones are more likely to have more RF energy. Hence, the harvesting circuit needs to choose the optimal frequency band and time, which would be achieved by using machine learning to predict where and when the most amount of RF energy would be present.

The ability to predict RF energy levels is crucial in energy harvesting applications as it can help to minimize the power consumption of the energy harvesting circuit. By analyzing the predicted RF energy, the circuit can determine whether the amount of RF energy available is worth keeping the circuit on or if it should be in sleep mode and save power (Mekid et al., 2017). Furthermore, in cases where the harvested energy is insufficient to power the load, predicting this event can enable the system to take appropriate actions, such as wireless power transfer or placing the load in sleep mode (Kaushik et al., 2015), (Ma et al., 2021). This is especially useful in WSNs where the nodes are powered periodically. In some applications, predicting the RF energy is essential to help select the optimum frequency channel for harvesting (Hooshiary et al., 2018).

Machine learning has been applied in many applications of communication systems. In RFEH circuits, some studies developed the optimal harvesting policies in RF energy harvesting devices. In (Hoang et al., 2014), the Markov decision process (MDP) along with an online algorithm was proposed to obtain the best channel access strategy for transmitting data or harvesting energy in cognitive radio networks. Also, MDP was used to decide the optimal power allocation in energy harvesting-powered devices (Li et al., 2019), (Xu et al., 2020).

Some works presented learning techniques to choose the optimal location or frequency band for harvesting. A decision policy in (Darak et al., 2016) using the Bayesian multi-armed bandit (MAB) method was developed to select the optimal sub-band for harvesting. Authors in (Kwan et al., 2020) presented a protocol that enables sensors in WSNs to harvest RF energy from intended and unintended sources. They use two algorithms, a linear forecaster provided with a linear regression-based enhancer and artificial neural networks, to decide the optimal schedule for RFEH. Researchers in (Yao and Ansari, 2021) use the RFEH technology is used to charge drones' batteries. They present an energy harvesting policy to minimize the long-term power consumption of drones. None of these works employed machine learning methods for modelling the RF signal itself.

In the wireless-powered communication network (WPCN), a learning algorithm based on Bayes' theorem was used in hybrid access point (HAP) to estimate the amount of energy consumed by wireless devices that harvest energy from HAP (Abuzainab et al., 2017). Researchers in (Munir and Dyo, 2018) considered the human mobility effect on the storage medium, for example, in the case of wearable devices. They proposed a predictor based on the Kalman filter to estimate the available energy and exchange between two different-sized capacitors based on the surrounding conditions. In (Koirala et al., 2019), a prediction algorithm based on moving average was proposed to predict energy in a WSN node by considering the history of this node as well as all the surrounding ones. Although these works addressed the issue of optimizing the RF energy harvesting process, these works did not use actual RF measurements.

Two algorithms, linear regression, and decision trees were proposed in (Azmat et al., 2016) to forecast RF energy for a given time and frequency. In (Eid et al., 2019), support vector machines were presented to predict the maximum obtainable power from Wi-Fi in five different locations. In (Ye et al., 2021), Four machine learning techniques were proposed to predict the available RF energy in communication systems. Very few works of literature have considered the prediction of RF energy as time series data. Moreover, none of these works addressed the effect of outliers on the model performance, which can lead to inaccurate results, and also the effect of different days of the week on the harvesting efficiency and model performance. These studies haven't discussed the prediction of the RF signal for a time interval. Long-term predictions require more powerful and complex learning techniques.

Motivated by the above insights, this paper suggests a complete artificial intelligence (AI) workflow for RF energy modelling. We employ powerful ML methods to pre-process and model the real-time RF energy data. This work aims to find the optimal predictive model for RF energy by exploring and comparing different time series algorithms. We model the RF energy as time-series data and present four machine learning techniques to forecast the mean value of the RF energy that can be harvested. Our evaluation of the performance of the time series models reveals that long short-term memory (LSTM) outperforms other techniques by achieving 95.76% prediction accuracy and that DeepAR is the most stable model with a 0.0212 range of errors. The key contributions of this work can be presented as follows.

- We measure the RF signal at different frequency bands using a software-defined radio (SDR) system.
- The Random Cut Forest (RCF) algorithm is used in pre-processing the data to eliminate the outliers.
- We optimize the LSTM model by modifying the loss function so that it eliminates the anomalies effect and hence, the model no longer needs the anomaly detection algorithm in the inference stage.
- The model is designed to be applicable for all days, and we evaluate the model performance for different days of the week.

The remainder of this paper is structured as follows. In section 2, RF data measurement will be introduced. The presented pipeline will be illustrated in section 3. Data pre-processing, selection of important parameters, a brief description of the time-series models, and the proposed optimized model are explained in subsections 3.1 through 3.7. Section 4 discusses the obtained results for predicting the RF energy using different models, and section 5 gives some conclusions on the research.

2. RF data acquisition

SDR is a radio whose hardware components can be implemented and controlled by software (Akeela and Dezfouli, 2018). This allows the radio to be more adaptable to different functionalities without the need to change the hardware architecture (Molla et al., 2022). SDR has been used in various military and non-military applications, including Mobile Communications, Cognitive Radio, and Wireless Sensor Networks.

The block diagram shown in Fig. 1 represents the SDR receiver where the RF waves are captured via an antenna and then converted to Intermediate Frequency (IF) through the RF Front End (RFFE) (Pozniak et al., 2019). Next, the Analog to analog-to-digital converter (ADC) converts the wave to a digital form for further processing. After that, the Digital

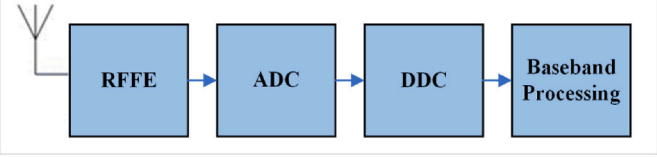

Fig. 1. SDR receiver block diagram.

Down Converter (DDC) translates the digital IF signal into baseband samples centered at zero frequency with a lower sampling rate. The signal can then be processed by software.

USRP N210 is the hardware used here as an interface between the RF spectrum and the software. USRP was designed by National Instruments to capture RF signals via an antenna connected to its RF port (Ettus Research). The USRP contains a Spartan 3 A-DSP 3400 FPGA, a 14-bit ADC that can sample at a rate up to 100 MS/s, a 16-bit DAC with 400 MS/s, and the WBX daughterboard. In this experiment, An 850–6500 MHz PCB log periodic antenna is used to receive the RF signal. This USRP is connected to the host PC through an ethernet interface.

The software platform used for baseband processing is GNU Radio, an open-source development framework used for the implementation of SDR (GNU Radio). It contains a variety of signal processing blocks. An application is implemented by connecting blocks that exist in gnu radio to compose a flowgraph.

The signal is measured using the conventional energy detection technique shown in Fig. 2. This method depends on passing the signal through a band pass filter (BPF) of a specified bandwidth followed by a squaring device, then an integrator to integrate the signal over a time interval (Shukla et al., 2016).

3. Experimental methods

In the approach outlined in this paper for model generation, we follow multiple stages, as illustrated in Fig. 3. The first stage is collecting the raw data, which is the RF energy signal. We use the SDR technology to collect the RF data. We utilize the universal software radio peripheral (USRP) N210 as our hardware in this implementation, along with the GNU radio software. The second phase is data preparation which includes data standardization and data cleaning. Standardization involves computing the required standard deviation and mean values from the dataset. For data cleaning, we use an anomaly detection algorithm to detect the outliers in the data. The output of this phase is the data that can be used in the following stages for model training and testing.

The processed data are trained using four algorithms: LSTM, DeepAR, Prophet, and Autoregressive Integrated Moving Average (ARIMA). This involves optimizing the model parameters to minimize the error between the predicted outputs and the actual outputs. Once the models are trained, they are evaluated using a separate dataset that was not used in the training process. This helps to assess the model's performance and identify any potential issues such as overfitting or underfitting. Eventually, based on the highest level of performance attained during the evaluation phase, one of these algorithms is chosen for making predictions.

3.1. Data collection

The RF signal was measured for three months in 6 cellular frequency bands which include 880–915 MHz, 925–960 MHz, 1710–1785 MHz, 1805–1880 MHz, 1920–1980 MHz, and 2110–2170 MHz. Each band is divided into frequency bins of 0.2 MHz bandwidth. Data points of the same frequency bin are captured every 3 s. For models' evaluation, we compute the normalized root mean square error (NRMSE) for N number of time samples, expressed by Eq. (1) and Eq. (2), to represent the prediction error. Each model was trained and evaluated using the same day's data in the same frequency band.

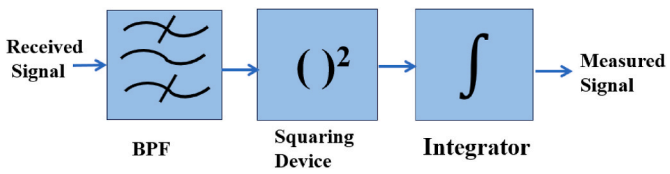

Fig. 2. Conventional energy detector.

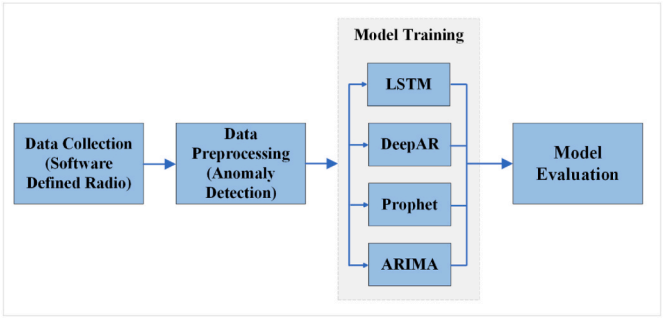

Fig. 3. ML workflow for model generation.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Actual\ data\ point_i - Predicted\ data\ point_i)^2}{N}} \quad (1)$$

$$NRMSE = \frac{RMSE}{max(Actual\ data) - min\ (Actual\ data)} \quad (2)$$

3.2. Data pre-processing

Before modelling, our first step is to ensure the data is clean and contains no outliers. Outliers are observations that diverge from the data pattern and have a negative impact on the model performance. We used the Random Cut Forest (RCF) algorithm to detect outliers in the dataset. RCF identifies an anomaly score for each point in the dataset, as shown in Fig. 4. Low scores imply that these data points are considered normal points, on the other hand, high scores indicate anomaly data points. From Fig. 5, detecting and handling anomalies increases the prediction accuracy. In our experiment, it was observed that the prediction error decreases by at least 2.78% in chunk seven after handling the outliers. The average value of NRMSE of the eight chunks containing outliers is 0.1169, whereas the average value of NRMSE without the outliers is 0.069.

3.3. Selecting the number of observations

The number of observations refers to the number of samples used for model training and testing. Different numbers of samples, starting from 1000 to 20,000 samples, have been examined in this work to select the optimal number of observations.

In Fig. 6, we evaluate the model performance using 2000, 3000, 4000, and 5000 observations. When we use a dataset of 2000 samples, the lowest value of error captured is 0.0389 in chunk 7, whereas the highest error value is 0.0819 in chunk 4. The average value of the NRMSE using 2000 samples is 0.0545. When using 3000 samples, the best prediction value comes out in chunk 2 with an NRMSE of 0.0373, whereas chunk 7 records the highest prediction error as 0.0732; this gives an average error value of 0.0487.

When using 4000 samples, the lowest prediction error arises in

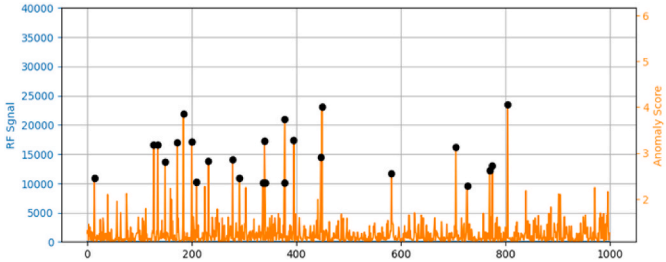

Fig. 4. Anomalies in RF energy data.

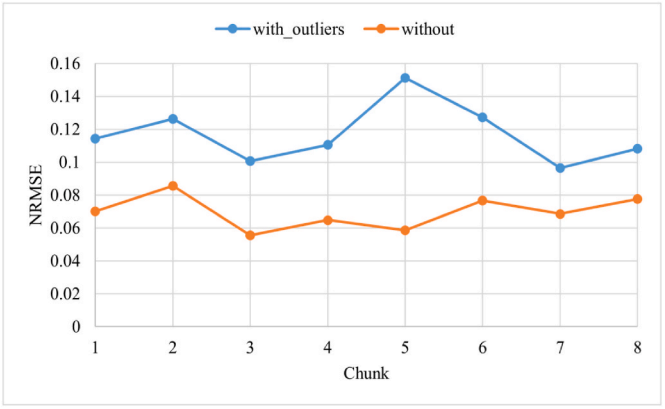

Fig. 5. NRMSE of different chunks of data in the LSTM model with and without outliers.

chunk 3 with an error of 0.0351; on the other hand, the highest prediction error is observed in chunk 5 with an error of 0.0623. The mean prediction error when using 4000 samples is 0.0464. When a dataset of 5000 samples is applied, the lowest prediction error is observed in chunk 4 with a value of 0.0337. The highest prediction error is recorded as 0.0591 in chunk 7, which results in an average error of 0.0424. The results show that the model's performance is best when using 5000 samples of observations.

Fig. 7 shows the mean NRMSE of different chunks for different numbers of observations. Numbers of observations fewer than 2000 have shown poor performance, whereas larger data sets have shown little or no improvement compared with the performance of datasets of less than 5000 samples, causing the model to consume much more time.

3.4. Window size selection

In this subsection, we study the impact of different window sizes on the model accuracy. Window size is the number of samples taken as history before predicting the next sample. Different window sizes have been tested to determine the optimal choice of window size. In predicting the mean of a time interval, it has been observed that when we increase the window size, the training time of the model overgrows, causing the model to be very slow. On the other hand, the prediction accuracy records slight improvements as the window size increases. A window size of 30 has been selected for predicting the mean of 100 samples, which means that we only need the samples of the last minute and a half to predict the average RF energy of the next 5 min. As shown in Fig. 8, the NRMSE reduces rapidly for window sizes less than 10, then larger sizes achieve a slight change in the NRMSE.

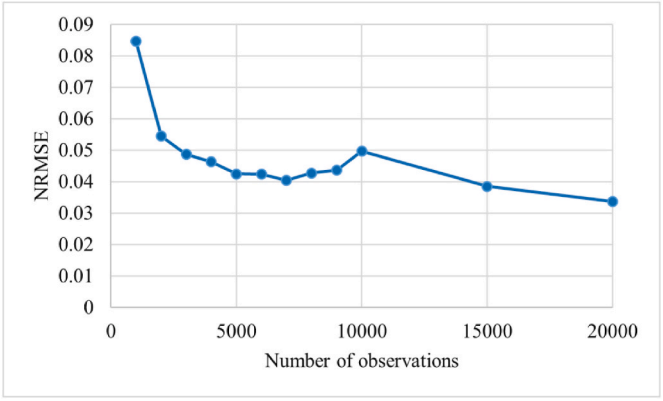

Fig. 7. The mean NRMSE of different numbers of observations for the LSTM model.

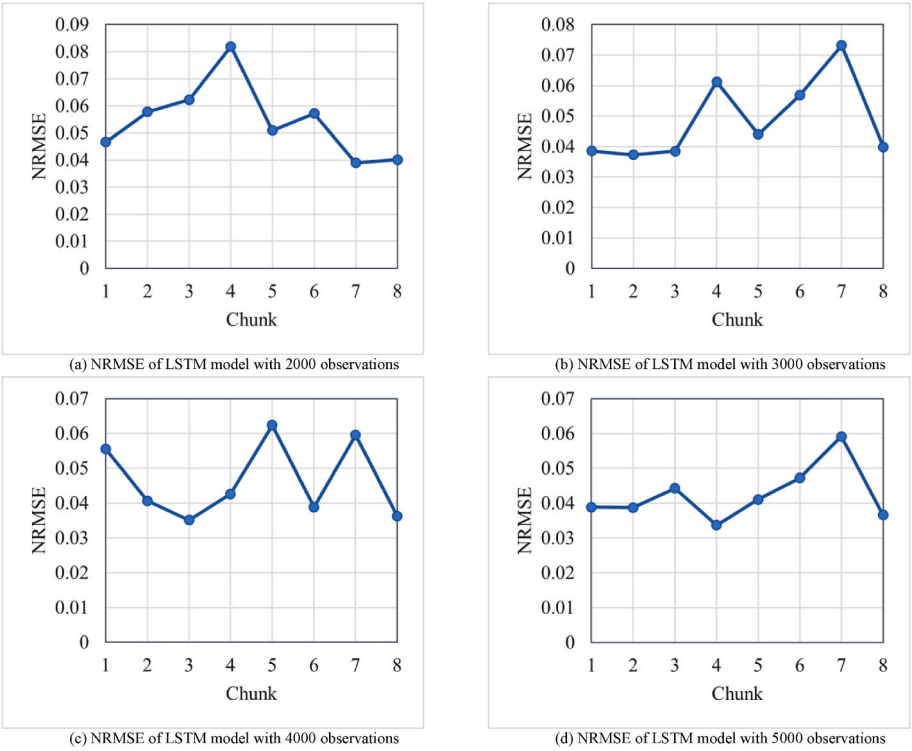

Fig. 6. NRMSE for LSTM model in 8 chunks of data with window size = 30, future interval = 100, and different observations of (a) 2000 observations (b) 3000 observations (c) 4000 observations (d) 5000 observations.

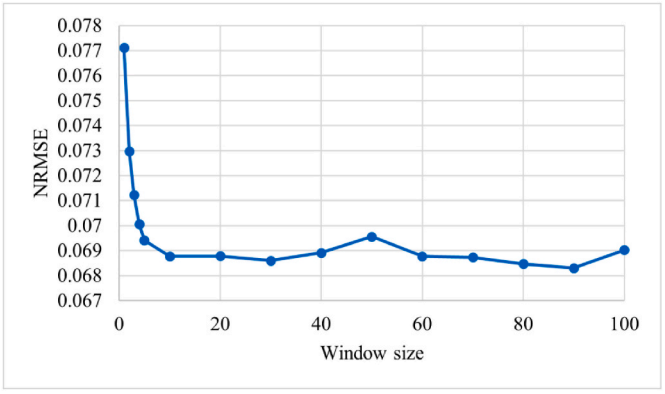

Fig. 8. Prediction error of different window sizes for predicting the mean of 100 samples using 5000 observations in 1960 MHz frequency.

3.5. Future interval selection

The future interval is the time interval in the future of which we predict the mean. Hence, we study this interval's length or its number of samples. Experiments have shown that the model can forecast the mean of intervals up to 5000 samples with almost the same average prediction accuracy. But as we increase the time interval, the mean of the intervals tends to be a constant value compared to the original instantaneous values, so the model predicts the average of the whole data set, not the average of a specific period. From Fig. 9, the mean of intervals larger than 300 samples, representing 15 min, are more likely to converge to the mean value of all data. In contrast, intervals of samples less than 300 have variations across different time intervals. Hence, the mean RF energy of the next 15 min can be accurately predicted.

3.6. Time series models

As mentioned before, we model our RF energy data as time series data since they are a collection of time-related observations. Time series prediction relies on analyzing current and past observations and developing a model to predict future observations. LSTM, DeepAR, Prophet, and ARIMA are presented in this research. The four techniques are evaluated and compared with respect to their error rates. This subsection gives a brief description of each of these algorithms.

The LSTM network can be defined as a particular category of Recurrent Neural Networks with the ability to remember the sequence of the data (Sherstinsky, 2020). In LSTM architecture, the previous stage's output is applied as the input to the current stage. The network decides at each step whether to modify the data in its memory or not; hence it keeps track of the valuable information in every step (Benhaddi and Ouarzazi, 2021). LSTM performs well with sequence data. One key advantage of LSTM is that it helps to overcome the problem of vanishing gradients and hence can capture long-term dependencies in the sequences. On the other hand, LSTMs are prone to overfitting and may require more memory to train.

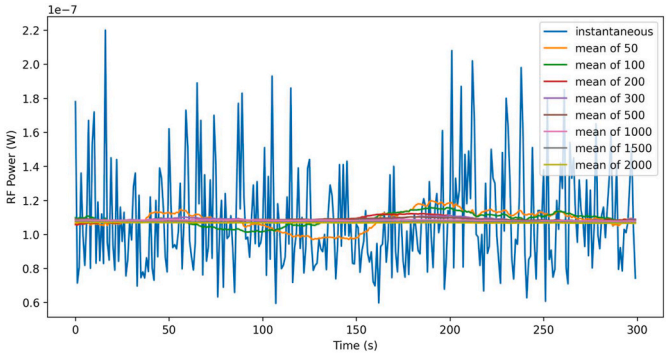

Fig. 9. The mean RF energy value for different numbers of samples.

The proposed LSTM model architecture, presented in Fig. 10, comprises an input layer, three LSTM layers of 32, 16, and 8 hidden units, and an output layer. The first two hidden layers have a sequence-to-sequence architecture, whereas the third has a sequence-to-label architecture. The rectified linear unit (ReLU) activation function is selected in all layers, excluding the output layer, which has linear activation since we need to predict a continuous numerical value. The adaptive moment (ADAM) optimization algorithm is used due to its computational efficiency, low memory usage, and its ability to handle large datasets. The model uses a value of 10^-3 as the learning rate and the mean square error for computing the loss function. The grid search optimization technique has been used to tune the hyper-parameters of the model. It searches through a defined set of values for each hyper-parameter to get the optimal values of the hyper-parameters that achieve the best performance.

DeepAR is a deep learning algorithm proposed by Amazon and used for forecasting time series data. DeepAR uses autoregressive recurrent neural networks with LSTM cells in the architecture (Salinas et al., 2020). DeepAR can have a more complex architecture than LSTM; however, it has many technical benefits. One remarkable advantage of DeepAR over traditional methods is that it can learn from multiple data series. Therefore, it can capture seasonality and complex relationships in the data with minimum feature modification. Furthermore, this method can predict the probability distribution of the sequence data series. It can also predict time series with little history samples. As is the case with LSTM, there are a few hyperparameters that can be tuned to best affect the algorithm accuracy in this model. We used three hidden layers, each of 32 cells, with a learning rate of 10^-3 and a batch size of 64.

Prophet, a forecasting algorithm released by researchers at Facebook, is a procedure for predicting time series data (Taylor and Letham, 2018). It was developed to intuitively adjust the parameters without knowing details about the underlying model. This algorithm is based on an additive model of three components. The trend component represents non-regular changes in the data. The seasonality component represents the data's periodic changes (daily, weekly, or annual). The third component observes the holiday effect. One advantage of Prophet is that it can model multiple seasonality and can integrate holidays into the model. In addition, it can predict time series data with missing dates. On the other hand, it can perform poorly with low numbers of observations.

The Autoregressive Integrated Moving Average is considered one of the most popular time series models that can be used with non-stationary time series data (Siami-Namini et al., 2019). ARIMA model is a general form of Autoregressive Moving Average (ARMA), and it combines three processes. The autoregression process uses past observations to predict the current one, the moving average process takes into consideration the lagged errors, and the integrated process converts non-stationary time series into stationary. The ARIMA model presumes that the data values have a linear correlation structure, and hence it is better used in predicting one sample in the future. One potential advantage of ARIMA is that it only needs the prior data of the time series to generalize the model. It performs well with short-term predictions. On

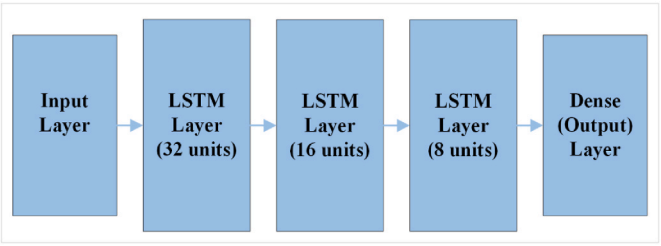

Fig. 10. Proposed LSTM architecture.

the other hand, it performs poorly with long-term predictions, it isn't used with seasonal time series data, and it performs poorly with turning points.

3.7. Optimized proposed model

The loss function is a significant component of the accuracy of the model. To improve it, a modified Mean Square Error (MSE) loss function is used in the proposed model. The purpose of the optimized model is to eliminate the effect of outliers in the data and hence improve the training process as well as the performance of the model. The traditional MSE loss is calculated by Eq. (3)

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3}$$

Where N is the number of training samples, y is actual output, and $\hat{y}$ is the predicted output.

The proposed model takes advantage of the anomaly scores obtained in the preprocessing stage. Since each data point is assigned a score value to identify the anomalies, we calculate the probability P of each data point being an anomaly in Eq. (4). This value describes how much each data point deviates from the entire dataset.

$$P_i = \frac{score_i - score_{min}}{score_{max} - score_{min}} \tag{4}$$

$$Loss_{Optimized} = \frac{1}{N}\sum_{i=1}^{N}((y_i - \hat{y}_i) * (1 - P_i))^2 \tag{5}$$

Enhanced loss function in our proposed solution is obtained by Eq. (5). The error resulting from outlier data points is then reduced using this number P, while the error resulting from typical data points is maintained. The proposed solution improves the accuracy of the model by minimizing the prediction error.

4. Results and discussion

4.1. Performance comparison of the 4 algorithms

In this experiment, the 1960.1 MHz frequency bin is utilized for evaluation. Performance results, presented in Table 1, describe the NRMSE for predicting RF energy samples using the four algorithms. Each value in the table represents the average error value of using eight chunks in each method. Each chunk has 5000 samples of observations. We predict the mean of the following 100 samples, which enables the harvesting circuit to decide to harvest once every 5 min instead of every 3 s. LSTM achieves the best performance with an average NRMSE of 4.24%, followed by DeepAR with a 4.98% prediction error, then the Prophet algorithm with an average error of 5.11%. The ARIMA model recorded the lowest prediction accuracy with a 5.75% error.

Fig. 11 shows the NRMSE of eight chunks of data in the four algorithms. The best performance of LSTM is observed in chunk 4 with an accuracy of 96.63%, and the worst performance is recorded in chunk 7 with 94.09% prediction accuracy. The lowest NRMSE of DeepAR takes place in the fourth chunk, with a prediction accuracy of 95.96%, whereas the seventh chunk records the lowest accuracy at 93.84%. In

Table 1
Performance comparison of LSTM, DeepAR, Prophet, and ARIMA in terms of NRMSE.

ML Algorithms	Mean NRMSE in 1960.1 MHz frequency
LSTM	0.0424
DeepAR	0.0498
Prophet	0.0511
ARIMA	0.0575

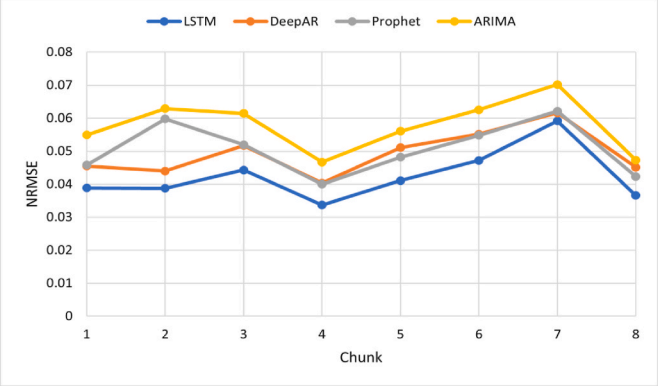

Fig. 11. NRMSE of the four algorithms in 8 different chunks with window size = 30, future interval = 100, and 5000 observations.

Prophet, the highest performance is recorded in chunk 4, with an accuracy of 96%, and the worst performance is 93.79% prediction accuracy in chunk 7. The highest prediction accuracy using ARIMA is 95.33% in chunk four, whereas the lowest prediction accuracy occurs in chunk 7 with 92.98% accuracy. The results indicate that LSTM performs better than other algorithms in all chunks in respect of NRMSE. However, it can be observed that DeepAR has the lowest variation of errors which makes DeepAR the most stable algorithm. DeepAR records a range of errors of 0.0212, followed by Prophet with 0.0222, whereas LSTM records the highest range of errors as 0.0255, and the range of errors in ARIMA is recorded as 0.0235. Actual RF energy samples and predictions of the four algorithms in the first chunk are shown in Fig. 12.

Predicting the mean value reduces the number of harvesting decisions the circuit needs to take and the number of on/off the circuit makes. In addition, mean prediction is more accurate than instantaneous value prediction. Fig. 13 presents the accuracy of the four algorithms when predicting one sample and when predicting the mean of a period of samples. Results indicate that the prediction accuracy for predicting the samples' mean of a time interval is higher than predicting one sample.

4.2. Performance of the proposed enhanced model

The improved loss function was evaluated in comparison to the conventional loss function for LSTM model while utilizing data that included outliers. The proposed method achieved lower error values compared to the traditional method for different datasets. The results are presented as normalized root mean squared error (NRMSE) values in Table 2.

Based on the provided results, it appears that the enhanced model generally outperforms the LSTM model in terms of prediction error. Across all datasets, the enhanced model has lower error rates compared

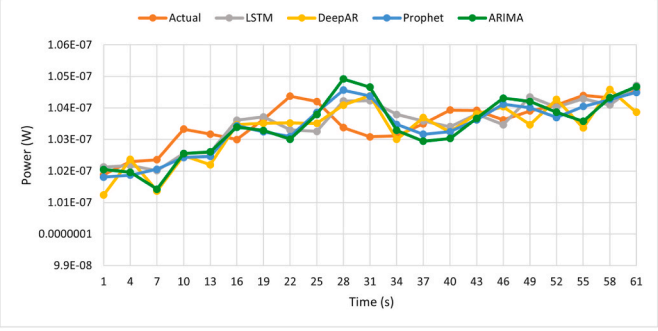

Fig. 12. Predicted RF energy using the four algorithms in 1960.1 MHz frequency.

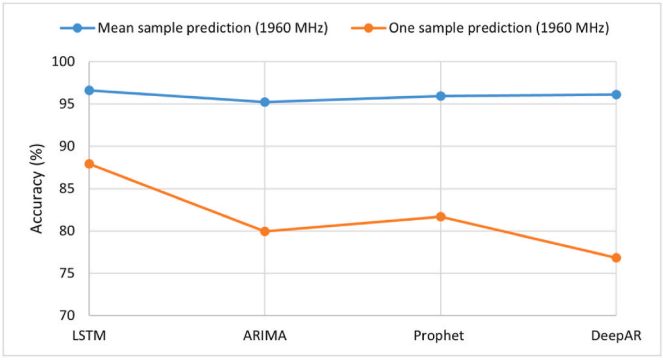

Fig. 13. Prediction accuracy comparison between one sample and mean sample prediction.

Table 2
Performance comparison of LSTM and proposed model in terms of NRMSE.

	LSTM	Enhanced Model
Dataset 1	0.1305	0.0937
Dataset 2	0.1154	0.0843
Dataset 3	0.2302	0.1182
Dataset 4	0.2066	0.109
Dataset 5	0.1966	0.1023
Dataset 6	0.2496	0.1283
Dataset 7	0.0831	0.0823
Dataset 8	0.0693	0.069
Dataset 9	0.0831	0.0823
Dataset 10	0.0798	0.0774

to the LSTM model. However, it is worth noting that the performance gain of the enhanced model over the LSTM model varies across different datasets. For example, in Dataset 9, the performance gain is relatively small, while in Dataset 3, the enhanced model significantly outperforms the LSTM model. The reason for this variation is attributed to the characteristics of the datasets, particularly the quantity and degree of deviation of the anomalies from the normal data points. Specifically, datasets 1 to 6 contain a larger number of high-level anomalies, whereas datasets 6 to 10 exhibit fewer and less severe anomalies. In general, the average error value for the traditional model is 0.1407, while the enhanced model records a 0.0954 average prediction error. However, this model should be trained on different datasets to ensure high performance in making predictions.

4.3. Analysis of energy prediction and harvester accuracy

Fig. 14 represents a Box-plot diagram of the NRMSE distribution in different chunks of data in one day using the four algorithms. It can be observed that the minimum and median error values in Prophet and DeepAR are almost the same, but DeepAR provides a lower interquartile range of error values. LSTM presents the lowest values of NRMSE, while Prophet presents the highest.

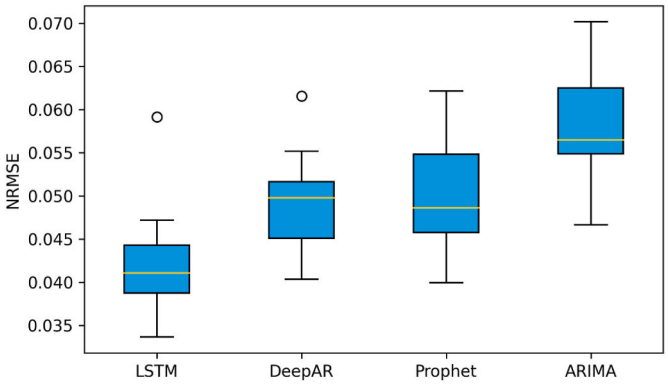

Fig. 14. Boxplot of NRMSE distribution of the four algorithms.

The predictive models can be used to increase the efficiency of the harvesting process. If the predicted energy value is lower than the threshold, the harvester will go to sleep mode. If the predicted energy value is higher than the threshold, the harvester will harvest the RF energy.

A false positive decision occurs when the actual energy value falls below the threshold, and the predicted energy value is above the threshold. A false negative decision occurs if the real value is above the threshold and the predicted value is below it. In these cases, the harvester will make a wrong harvesting decision. We evaluated the harvester decision efficiency using precision and recall evaluation measures. Precision, also known as the Positive Predictive Value (PPV), measures the proportion of values predicted above the threshold and are correctly above the threshold. Recall or True Positive Rate (TPR) measures the proportion of real values above the threshold that are correctly predicted above the threshold.

In Fig. 15, we use the LSTM model to evaluate the harvester efficiency per day with a threshold set to 0.11 μW. In this experiment, the mean number of false positives is 12, and the mean number of false negatives is 9.

The model achieves average precision and recall values of 82.01% and 80.68%, respectively. In Fig. 16, we use the DeepAR model with the same threshold. The mean number of false positives is 12.13, while the mean number of false negatives is 14.63. The DeepAR model records 70.99% and 70.35% average precision and recall values, respectively.

To discuss the applicability of the model for all days and evaluate its performance across different days of the week, the LSTM model was tested and evaluated across datasets from all days of the week. Fig. 17 presents the harvester accuracy distribution per day for the LSTM model. The error rate for some days is more variable than others; however, the median value of accuracy oscillates between 62.5% and 92.9%. It can be observed that working days have higher chances of fault harvester decisions than weekends (Friday and Saturday). This can be explained that weekends have more stable energy signals during the day, unlike working days.

4.4. Analysis of the model complexity

This section analyses the computational complexity of the models represented as a function of parameters related to both the data and the technique, depending on how it is implemented. For LSTM, the time complexity per weight for each time step is $O(1)$ (Tsironi et al., 2017). As a result, the time complexity of an LSTM is $O(W)$, where W is the number of weights. Therefore, the time complexity of the LSTM model is $O(ij + jk + kl + lm)$ where $i, j, k, l,$ and m stand for the number of nodes in the input layer, the second layer, the third layer, the fourth layer, and the output layer, respectively. The computational complexity of DeepAR can be approximated as the same complexity of LSTM since it is an LSTM-based recurrent neural network.

For the ARIMA model, the complexity is calculated as $O(n^2T)$ where n is the total number of parameters p, d, and q and T is the size of the time series (Wang et al., 2022). It should be emphasized that the complexity of the ARIMA model increases quadratically. However, the number of nodes in LSTM architecture makes LSTM and DeepAR more computationally expensive.

On the other hand, the computational complexity of neural networks can be reduced through various techniques (Masuko, 2017). Pruning techniques can be used to remove unnecessary connections, nodes, or weights from the neural network, which can significantly reduce computational complexity. Quantization techniques can be used to reduce the precision of weights and activations in the neural network, which can reduce the amount of memory and computational resources

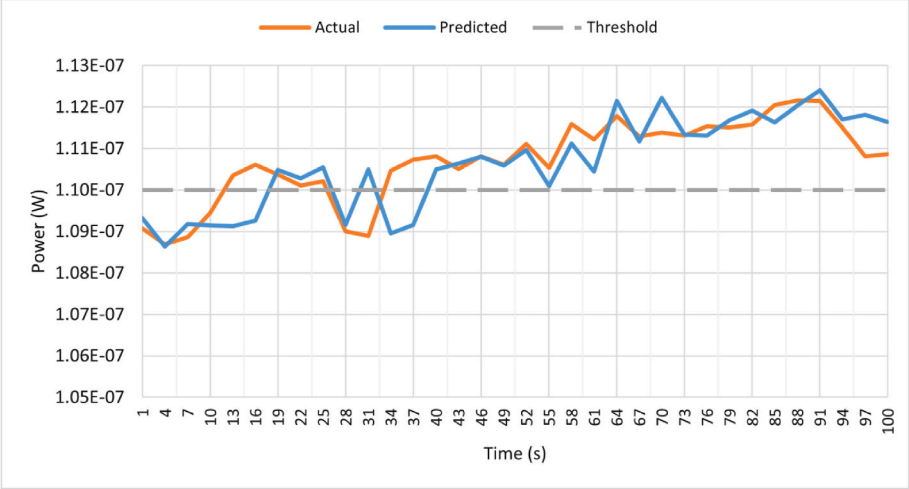

Fig. 15. Efficiency test for energy prediction using LSTM with 0.11 μW threshold.

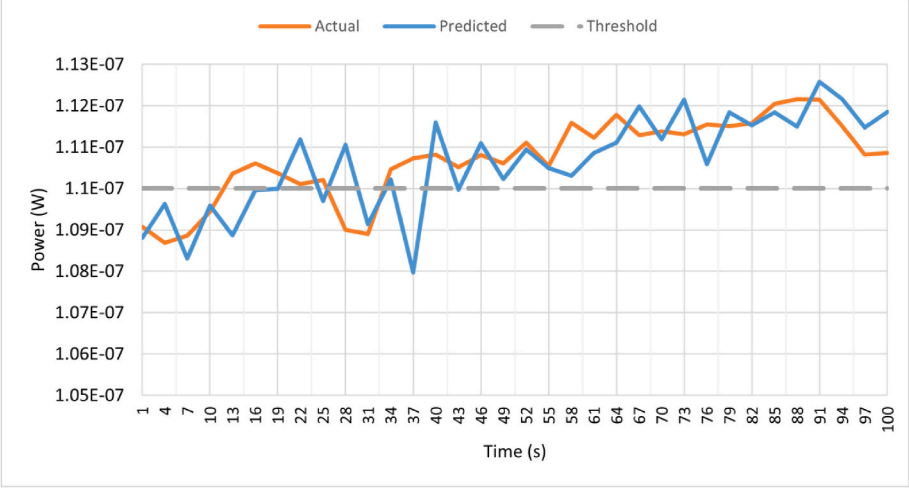

Fig. 16. Efficiency test for energy prediction using DeepAR with 0.11 μW threshold.

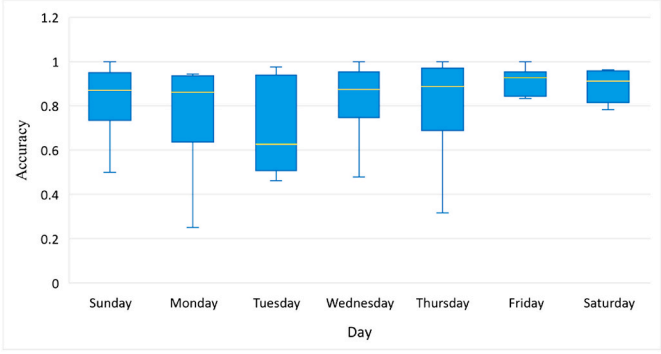

Fig. 17. Boxplot representation of harvester accuracy distribution per day.

required. In addition, various compression techniques, such as weight sharing and knowledge distillation, can be used to reduce the size and computational complexity of neural networks.

4.5. Performance comparison with the state-of-the-art models

Given that LSTM has demonstrated superior performance compared to other techniques, this section presents a comparative analysis between LSTM and the most effective model among state-of-the-art models for predicting RF signals. Our literature analysis in Section 1 indicates that linear regression has the highest accuracy in the previous studies (Azmat et al., 2016; Eid et al., 2019; Ye et al., 2021). Therefore, we compare the performance of LSTM against LR.

In Fig. 18, a comparison between the models in terms of NRMSE is presented where we predict the next sample, i.e., the immediate succeeding sample. Based on the provided results, it appears that the LSTM model outperforms the LR model in terms of NRMSE for predicting the next sample.

Specifically, for all six chunks considered, the LSTM model achieved lower NRMSE values compared to the LR model. The difference between the NRMSE values of the two models varies across the different instances. However, the difference between the NRMSE suggests that the LSTM model is significantly better than the LR model.

In Fig. 19, we evaluate the performance of predicting the mean of the next 100 samples, i.e., the mean of the next 5 min. For all seven instances considered, the LSTM model achieved lower mean prediction error values compared to the LR model. LSTM achieved an average prediction error of 0.0401 while LR achieved a 0.047 average prediction error.

Upon comparison of the models' robustness against outliers, it is observed that the proposed solution outperforms both the LR and LSTM

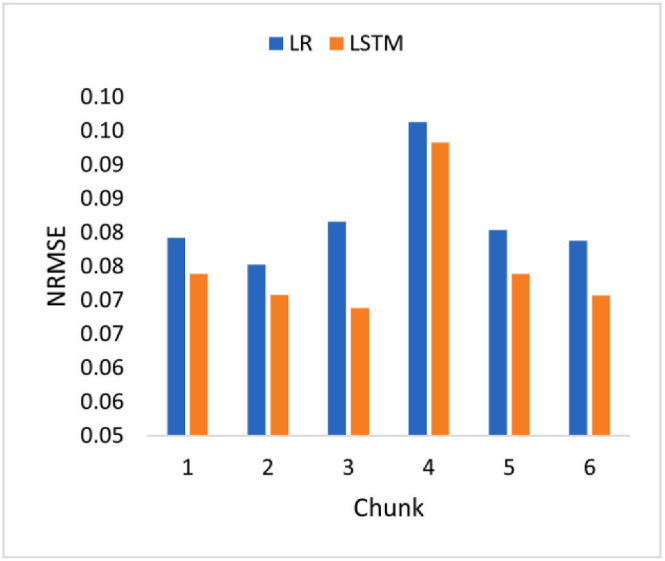

Fig. 18. Prediction error comparison between LR and LSTM for predicting the instantaneous values.

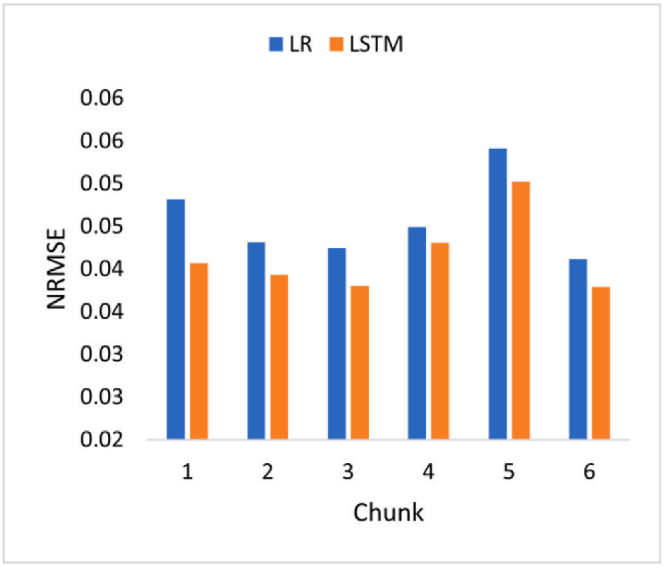

Fig. 19. Prediction error comparison between LR and LSTM for predicting the mean values.

models in terms of predicting the target variable, both with and without outliers. Fig. 20 indicates that when outliers are present, the proposed solution achieved an average NRMSE value of 0.0889, which is considerably lower than both the LR model's average NRMSE value of 0.1674 and the LSTM model's NRMSE value of 0.1185. This suggests that the proposed solution is better at handling outliers in the data and can produce more accurate predictions even in the presence of such anomalies. Moreover, the error of the LR model has a significant increase in the presence of anomaly data compared to the other two models. It can be concluded that LR is less robust to outliers than LSTM and the proposed model.

In terms of the data size, we evaluate the performance against different data sizes. The sample sizes used in previous works were 1200 and 3098 samples. Fig. 21 indicates that increasing the data sample size decreases the prediction error in both LR and LSTM.

Incorporating additional data into a machine learning model primarily affects the training process, specifically the number of floating-

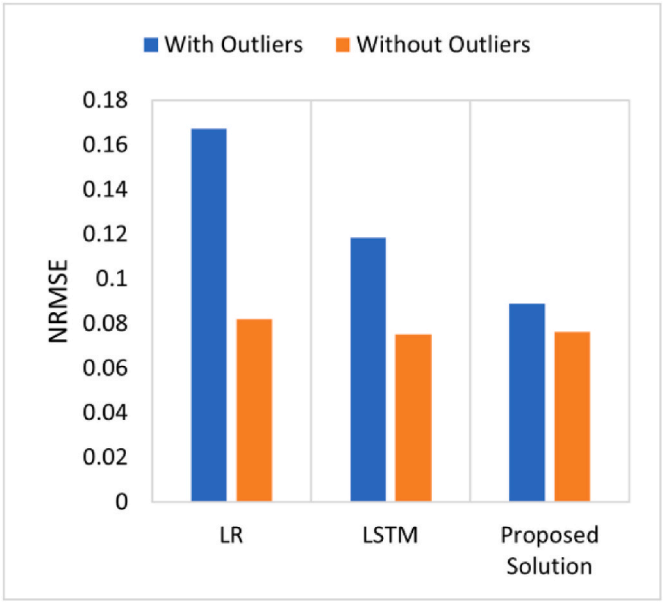

Fig. 20. Performance comparison between LR, LSTM, and the proposed model for datasets containing outliers.

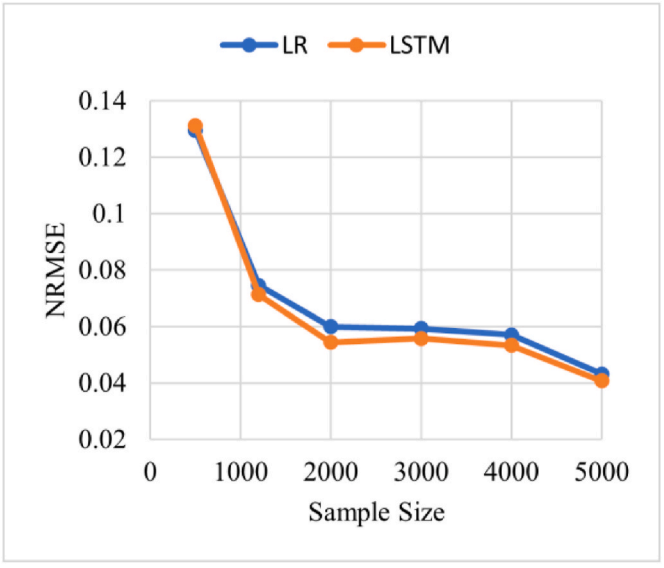

Fig. 21. NRMSE for different sizes of datasets in LR and LSTM.

point operations (FLOPs) required for training. However, it typically does not have a significant impact on the computational cost of making predictions. This is because the forward pass only involves applying the learned model parameters to the input data, which does not require additional training or adjustment of the model. Therefore, the computational cost of the forward pass is generally constant regardless of the amount of data used for training the model.

5. Conclusions

In this work, we propose a complete workflow for developing the optimal model to predict the energy of RF signals using various ML techniques. RF energy signal is measured at different frequency bands using SDR. Four different time series methods (LSTM, DeepAR, ARIMA, and Prophet) are discussed for comparison using the data of the 1960.1 MHz frequency bin. We predict the mean of the next 5 min to ensure that

the harvesting circuit consumes less power by making fewer harvesting decisions. The results indicate that LSTM has the highest accuracy of 95.76%, followed by DeepAR and Prophet. The ARIMA records the worst prediction accuracy, and DeepAR is the most stable algorithm. Additionally, this research proposes an optimized model based on an enhanced loss function to reduce the effect of anomalies on the model performance. The proposed optimized model achieves robustness against outliers, achieving 32.2% lower prediction error than the traditional model. To the best of our knowledge, there is currently no deep neural network model that can accurately predict RF energy or effectively deal with anomalies that may affect the model's performance in the field of RF energy harvesting. Accurately predicting RF energy is crucial for optimizing energy harvesting circuits and powering wireless and low-powered devices without relying on batteries.

It's critical to note that the suggested predictive model has some significant constraints such as having a relatively high computational complexity compared to other models. Given that the RFEH circuit has limited power, optimizing the algorithm so that it consumes minimum power is an essential consideration in future research. A future study can therefore explore quantization techniques for reducing the algorithm's size, weight pruning, and model compression for optimizing computational complexity. Moreover, we intend to incorporate additional baseline models, specifically those utilizing advanced deep learning architectures. This would allow for a more thorough comparison of the proposed model's performance against a diverse set of benchmarks.

Formatting of funding sources

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

CRediT authorship contribution statement

Shaimaa H. Mohammed: Conceptualization, Data curation, Formal analysis, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Ashraf S. Mohra:** Project administration, Supervision. **Ashraf Y. Hassan:** Project administration, Supervision. **Ahmed F. Elnokrashy:** Conceptualization, Formal analysis, Investigation, Project administration, Resources, Supervision, Writing – review & editing.

Declaration of competing interest

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

Data availability

Data will be made available on request.

References

Abuzainab, N., Saad, W., Maham, B., 2017. Robust Bayesian learning for wireless RF energy harvesting networks. In: 2017 15th Int. Symp. Model. Optim. Mobile, Ad Hoc, Wirel. Networks. WiOpt 2017. https://doi.org/10.23919/WIOPT.2017.7959919.

Akeela, R., Dezfouli, B., 2018. Software-defined radios: architecture, state-of-the-art, and challenges. Comput. Commun. 128 (March), 106–125. https://doi.org/10.1016/j.comcom.2018.07.012.

Azmat, F., Chen, Y., Stocks, N., 2016. Predictive modelling of RF energy for wireless powered communications. IEEE Commun. Lett. 20 (1), 173–176. https://doi.org/10.1109/LCOMM.2015.2497306.

Benhaddi, M., Ouarzazi, J., 2021. Multivariate time series forecasting with dilated residual convolutional neural networks for urban air quality prediction. Arabian J. Sci. Eng. 46 (4), 3423–3442. https://doi.org/10.1007/s13369-020-05109-x.

Darak, S.J., Moy, C., Palicot, J., Louet, Y., 2016. Smart decision making policy for faster harvesting from ambient RF sources in wireless sensor nodes. Proc. Int. Symp. Wirel. Commun. Syst. 2016 (Octob), 148–152. https://doi.org/10.1109/ISWCS.2016.7600891.

Do, H.D., Kim, D.E., Lam, M.B., Chung, W.Y., 2021. Self-powered food assessment system using LSTM network and 915 MHz RF energy harvesting. IEEE Access 9, 97444–97456. https://doi.org/10.1109/ACCESS.2021.3095271.

Eid, A., Mghabghab, S., Costantine, J., Awad, M., Tawk, Y., 2019. Support vector machines for scheduled harvesting of wi-fi signals. IEEE Antenn. Wireless Propag. Lett. 18 (11), 2277–2281. https://doi.org/10.1109/LAWP.2019.2943250.

Eltresy, N.A., et al., 2019. RF energy harvesting IoT system for museum ambience control with deep learning. Sensors 19 (20). https://doi.org/10.3390/s19204465.

Ettus Research." https://www.ettus.com/. (Last Accessed, March 2024).

GNU Radio." https://www.gnuradio.org/. (Last Accessed, March 2024).

Hesham, R., Soltan, A., Madian, A., 2021. Energy harvesting schemes for wearable devices. AEU - Int. J. Electron. Commun. 138 (July), 153888 https://doi.org/10.1016/j.aeue.2021.153888.

Hoang, D.T., Niyato, D., Wang, P., Kim, D.I., 2014. Opportunistic channel access and RF energy harvesting in cognitive radio networks. IEEE J. Sel. Area. Commun. 32 (11), 2039–2052. https://doi.org/10.1109/JSAC.2014.141108.

Hooshiary, A., Azmi, P., Mokari, N., Maleki, S., 2018. Optimal channel selection for simultaneous RF energy harvesting and data transmission in cognitive radio networks. Trans. Emerg. Telecommun. Technol. 29 (3) https://doi.org/10.1002/ett.3291.

Kaushik, K., et al., 2015. RF energy harvester-based wake-up receiver. In: 2015 IEEE SENSORS - Proc.. https://doi.org/10.1109/ICSENS.2015.7370562. March 2016.

Khan, D., et al., 2020. A 2.45 GHZ high efficiency CMOS RF energy harvester with adaptive path control. Electron 9 (7), 1–14. https://doi.org/10.3390/electronics9071107.

Koirala, B., Dahal, K., Keir, P., Chen, W., 2019. A multi-node energy prediction approach combined with optimum prediction interval for RF powered WSNS. Sensors 19 (24). https://doi.org/10.3390/s19245551.

Kwan, J.C., Chaulk, J.M., Fapojuwo, A.O., 2020. A coordinated ambient/dedicated radio frequency energy harvesting scheme using machine learning. IEEE Sensor. J. 20 (22), 13808–13823. https://doi.org/10.1109/JSEN.2020.3003931.

Lam, M.B., Nguyen, T.H., Chung, W.Y., 2020. Deep learning-based food quality estimation using radio frequency-powered sensor mote. IEEE Access 8, 88360–88371. https://doi.org/10.1109/ACCESS.2020.2993053.

Li, Q., Gao, J., Liang, H., Zhao, L., Tang, X., 2019. Optimal power allocation for wireless sensor powered by dedicated RF energy source. IEEE Trans. Veh. Technol. 68 (3), 2791–2801. https://doi.org/10.1109/TVT.2019.2892770.

Ma, Z., Yu, X., Guo, S., Zhang, Y., 2021. Analysis of wireless sensor networks with sleep mode and threshold activation. Wireless Network 27 (2), 1431–1443. https://doi.org/10.1007/s11276-020-02512-y.

T. Masuko, "Computational cost reduction of long short-term memory based on simultaneous compression of input and hidden state," 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Okinawa, Japan, 2017, pp. 126-133, doi: 10.1109/ASRU.2017.8268926.

Mekid, S., Qureshi, A., Baroudi, U., 2017. Energy harvesting from ambient radio frequency: is it worth it? Arabian J. Sci. Eng. 42 (7), 2673–2683. https://doi.org/10.1007/s13369-016-2308-y.

Molla, D.M., Badis, H., George, L., Berbineau, M., 2022. Software defined radio platforms for wireless technologies. IEEE Access 10, 26203–26229. https://doi.org/10.1109/ACCESS.2022.3154364.

Mouapi, A., Hakem, N., 2018. A new approach to design autonomouswireless sensor node based on RF energy harvesting system. Sensors 18 (no. 1). https://doi.org/10.3390/s18010133.

Munir, B., Dyo, V., 2018. On the impact of mobility on battery-less RF energy harvesting system performance. Sensors 18 (11). https://doi.org/10.3390/s18113597.

Olgun, U., Chen, C.C., Volakis, J.L., 2012. Design of an efficient ambient WiFi energy harvesting system. IET Microw., Antennas Propag. 6 (11), 1200–1206. https://doi.org/10.1049/iet-map.2012.0129.

Ozger, M., Cetinkaya, O., Akan, O.B., 2018. Energy harvesting cognitive radio networking for IoT-enabled smart grid. Mobile Network. Appl. 23 (4), 956–966. https://doi.org/10.1007/s11036-017-0961-3.

Powercast." https://eu.mouser.com/new/powercast/powercast-p2110-powerharvester-evb/. (Last Accessed, March 2024).

Pozniak, M., Sadhukhan, D., Ranganathan, P., 2019. RF exploitation and detection techniques using software defined radio: a survey. IEEE Int. Conf. Electro Inf. Technol. 2019 (May), 345–350. https://doi.org/10.1109/EIT.2019.8834203.

Salinas, D., Flunkert, V., Gasthaus, J., Januschowski, T., 2020. DeepAR: probabilistic forecasting with autoregressive recurrent networks. Int. J. Forecast. 36 (3), 1181–1191. https://doi.org/10.1016/j.ijforecast.2019.07.001.

Sherstinsky, A., 2020. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Phys. Nonlinear Phenom. 404, 132306 https://doi.org/10.1016/j.physd.2019.132306.

Shukla, S., Rao, A.K., Srivastava, N., 2016. A survey on energy detection schemes in cognitive radios. In: International Conference on Emerging Trends in Electrical, Electronics and Sustainable Energy Systems. ICETEESES 2016, pp. 223–228. https://doi.org/10.1109/ICETEESES.2016.7581389.

Siami-Namini, S., Tavakoli, N., Siami Namin, A., 2019. A comparison of ARIMA and LSTM in forecasting time series. In: Proc. - 17th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2018, pp. 1394–1401. https://doi.org/10.1109/ICMLA.2018.00227.

Sil, I., Mukherjee, S., Biswas, K., 2017. A review of energy harvesting technology and its potential applications. Environ. Earth Sci. Res. J. 4 (2), 33–38. https://doi.org/10.18280/eesrj.040202.

Syed, Y., Hegde, B.G., Prabhakar, T.V., Manjunath, M., Vinoy, K.J., 2017. RF energy harvesting chip powered sensor node. In: 2016 IEEE Int. Conf. Electron. Circuits Syst. ICECS 2016, pp. 748–751. https://doi.org/10.1109/ICECS.2016.7841310.

Taylor, S.J., Letham, B., 2018. Forecasting at scale. Am. Statistician 72 (1), 37–45. https://doi.org/10.1080/00031305.2017.1380080.

Tran, L.G., Cha, H.K., Park, W.T., 2017. RF power harvesting: a review on designing methodologies and applications. Micro Nano Syst. Lett. 5 (1) https://doi.org/10.1186/s40486-017-0051-0.

Tsironi, E., Barros, P., Weber, C., Wermter, S., 2017. An analysis of convolutional long short-term memory recurrent neural networks for gesture recognition. Neurocomputing 268, 76–86. https://doi.org/10.1016/j.neucom.2016.12.088.

Varghese, B., John, N.E., Sreelal, S., Gopal, K., 2016. Design and development of an RF energy harvesting wireless sensor node (EH-WSN) for aerospace applications. Procedia Comput. Sci. 93 (September), 230–237. https://doi.org/10.1016/j.procs.2016.07.205.

Vyas, R.J., Cook, B.B., Kawahara, Y., Tentzeris, M.M., 2013. E-WEHP: a batteryless embedded sensor-platform wirelessly powered from ambient digital-TV signals. IEEE Trans. Microw. Theor. Tech. 61 (6), 2491–2505. https://doi.org/10.1109/TMTT.2013.2258168.

Wang, X., Kang, Y., Hyndman, R.J., Li, F., 2022. Distributed ARIMA models for ultra-long time series. Int. J. Forecast. https://doi.org/10.1016/j.ijforecast.2022.05.001. Feng Li.

Xu, P., Flandre, D., Bol, D., 2019. Analysis, modeling, and design of a 2.45-GHz RF energy harvester for SWIPT IoT smart sensors. IEEE J. Solid State Circ. 54 (10), 2717–2729. https://doi.org/10.1109/JSSC.2019.2914581.

Xu, Y.H., Xie, J.W., Zhang, Y.G., Hua, M., Zhou, W., 2020. Reinforcement learning (Rl)-based energy efficient resource allocation for energy harvesting-powered wireless body area network. Sensors 20 (1), 1–22. https://doi.org/10.3390/s20010044.

Yao, J., Ansari, N., 2021. Wireless power and energy harvesting control in IoD by deep reinforcement learning. IEEE Trans. Green Commun. Netw. 5 (2), 980–989. https://doi.org/10.1109/TGCN.2021.3049500.

Ye, Y., Azmat, F., Adenopo, I., Chen, Y., Shi, R., 2021. RF energy modelling using machine learning for energy harvesting communications systems. Int. J. Commun. Syst. 34 (3), 1–14. https://doi.org/10.1002/dac.4688.